VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA BEZPEČNOSTNÍCH VLASTNOSTÍ V OS ANDROID

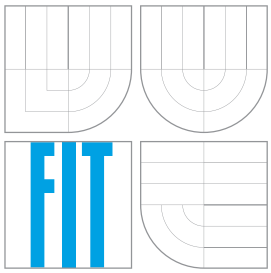
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN HANYÁŠ

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA BEZPEČNOSTNÍCH VLASTNOSTÍ V OS ANDROID

ANALYSIS OF SECURITY PROPERTIES IN OS ANDROID

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN HANYÁŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL OČENÁŠEK, Ph.D.

BRNO 2014

Abstrakt

Tato práce popisuje operační systém Android a jeho bezpečnostní aspekty. Dále se věnuje forenzní analýze tohoto operačního systému. Cílem práce je vytvořit forenzní aplikaci umožňující získat citlivá data a dále provést forenzní analýzu zavedenými nástroji a vytvořit podklady pro výuku.

Abstract

This thesis describes operating system Android and its security aspects. Furthermore, the thesis will focus on the forensics analysis of this operating system. The aim is to create forensics application which allows to get sensitive data as well as to make forensic analysis using established tools, and to create background materials for teaching.

Klíčová slova

Android, forenzní analýza, bezpečnost

Keywords

Android, forensic analysis, security

Citace

Martin Hanyáš: Analýza bezpečnostních vlastností v OS Android, diplomová práce, Brno, FIT VUT v Brně, 2014

Analýza bezpečnostních vlastností v OS Android

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Pavla Očenáška, Ph.D.
Uvedl jsem všechny prameny, ze kterých jsem čerpal.

.....
Martin Hanyáš
23. května 2014

Obsah

Kapitola 1

Úvod

Počet tzv. chytrých zařízení neustále roste – chytrých mobilních telefonů se prodává čím dál víc[1]. Někteří lidé si bez nich nedokáží představit ani krok, dalším lidem mohou nahradit, nebo doplnit pracovní nástroje. Tato a další fakta znamenají, že mobilní zařízení obsahují značné množství informací o svých vlastnících, či o lidech, se kterými jsou v kontaktu. Takováto data většinou nebývají nebezpečná, na druhou stranu mohou prozradit velmi mnoho informací o majiteli zařízení. Toho se hojně využívá při forenzní analýze, kdy ze zabavených či na místech činu nalezených zařízení forenzní specialisté získávají identifikaci pachatele nebo oběti.

V této diplomové práci se zabývám bezpečnostní analýzou a získáváním dat z operačního systému Android. Tento systém patří mezi nejrozšířenější mobilní platformy[2] a jeho open source povaha umožňuje podrobné zkoumání jeho fungování a vlastností.

Cílem práce je prozkoumat a popsat existující forenzní nástroje, vytvořit podklady pro výuku využívající tyto nástroje a dále navrhnout a implementovat aplikaci, která využívá známých vlastností operačního systému Android a s jejich pomocí umožňuje ze zkoumaného mobilního zařízení získat data, která mohou vést k zjištění informací o majiteli.

V prvních dvou kapitolách se zabývám teoretickým popisem digitální forenzní analýzy (se zaměřením na mobilní zařízení) a operačního systému Android. V dalších kapitolách se zabývám specifikací a návrhem aplikace. Následující kapitola se týká implementace aplikace, jejího uživatelského rozhraní, testování a použitých optimalizací, týkající se běhu na mobilním zařízení. Zde také popisuji postupy pro získávání dat z mobilního zařízení, které jsou v aplikaci použity. Sedmá kapitola popisuje postupy provedení forenzní analýzy již dostupnými nástroji. Tyto postupy jsou ve formě demonstračních úloh dostupné v přílohách. V závěru hodnotím dosažené výsledky a možná rozšíření.

[1] Podle společnosti IDC dokonce prodeje chytrých telefonů překonaly prodeje klasických telefonů: http://www.businesswire.com/news/home/20130425006953/en/Smartphones-Shipped-Q1-2013-Feature-Phones-Industry#.UtaYV7QeL80

[2] Statistiky zastoupení mobilních operačních systémů (květen 2014) podle StatCounter: http://gs.statcounter.com/#mobile+tablet-os-ww-monthly-201405-201405-bar

Kapitola 2

Digitální forenzní analýza

Digitální forenzní analýza je věda zabývající se ziskem dat z digitálních zařízení (počítače, mobilní telefony, GPS navigace...). Jedná se o poměrně nový vědní obor[1], který se rychle rozvíjí a stává se součástí prakticky každého vyšetřování. [8]

Stejně jako klasické forenzní vyšetřování, má i digitální forenzní analýza určité kroky prováděné během vyšetřování:

- Konzervace zařízení – zabránění či minimalizování možnosti provedení změn na zkoumaném zařízení (odpojení od sítě, zabránění vypnutí zařízení...).
- Zkoumání dat – extrahování dat ze zařízení a následná analýza (mimo zařízení, v laboratoři), hledání možných důkazů.
- Dokumentace – dokumentace průběhu získávání dat a samotných získaných dat.

Následná interpretace získaných a zdokumentovaných dat je už součástí práce vyšetřovatele. [11]

2.1 Forenzní analýza zařízení s OS Android

Při provádění forenzní analýzy mobilního zařízení s operačním systémem Android je problematický především první bod výše uvedeného seznamu. Důležitým krokem je odpojení zkoumaného přístroje od sítě, aby nemohlo dojít například k vzdálenému vymazání zařízení nebo změnám na základě změny polohy v mobilní síti. Přístroje s OS Android umožňují přepnutí do tzv. *Režimu letadlo*, pokud není možné přístroj přepnout okamžitě (například má zamčenou obrazovku), je možné jej umístit do některého ze speciálních obalů[2], které odizolují bezdrátové sítě. Následná forenzní analýza ovšem většinou vyžaduje mírný zásah do zařízení v podobě nainstalování drobné aplikace, která umožní propojení s forenzním softwarem běžícím na počítači (více v kapitole 2.1.1). Další z možných zásahů může být provedení *rootu* zařízení (viz kapitola 3.3), který umožní získání obrazu celé vnitřní paměti.

Další dva kroky už takové komplikace nepřinášejí, na trhu je několik různých nástrojů, které umožňují získávat data vcelku intuitivně i pro nezkušeného uživatele. I pro zkušeného forenzního technika představují zjednodušení a zpřehlednění práce. Některé vybrané nástroje budou představeny v následující podkapitole.

[1] V American Academy of Forensic Sciences byla sekce Digital and Multimedia Sciences, zabývající se digitální forenzní analýzou, vytvořena v roce 2008.

[2] Například výrobky společnosti Paraben – http://www.paraben.com/stronghold.html

2.1.1 Nástroje

V této kapitole představím několik vybraných nástrojů umožňujících forenzní analýzu zařízení s operačním systémem Android.

Hardwarové

UFED (Universal Forensic Extraction Device)

Jedná se o zařízení společnosti Cellebrite[3] uvedené v roce 2007. V současné době je k dispozici v několika variantách, lišících se nejen provedením, ale i vzhledem (na obrázku 2.4 je zobrazena varianta UFED Touch Ultimate). Jedná se o kombinaci hardware a specializovaného software, umožňující extrahování a analyzování dat z mobilních zařízení. [4] [6]

Obrázek 2.1: UFED Touch Ultimate. Zdroj: [6]

Softwarové

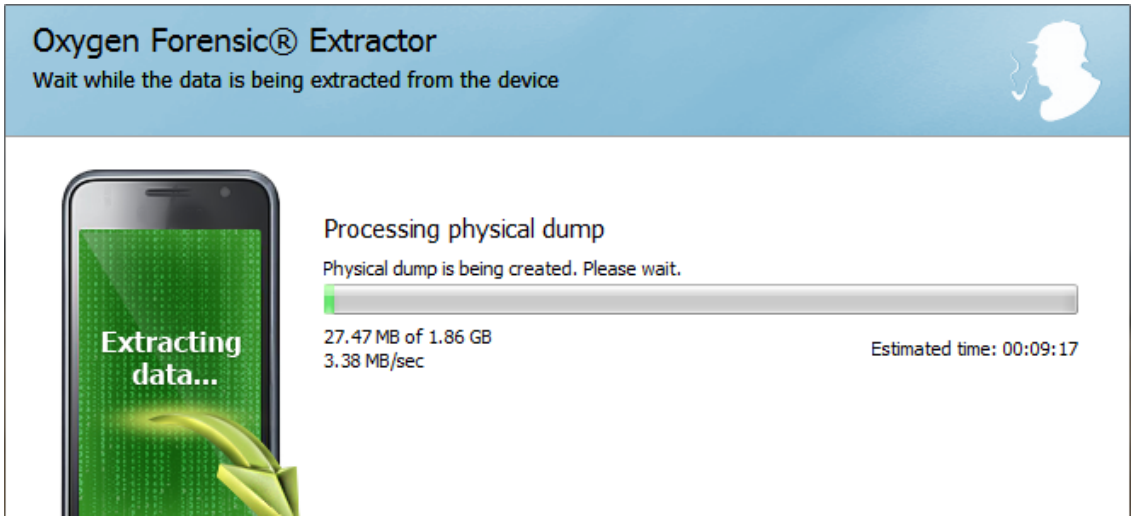
Nástrojů pro forenzní analýzu mobilních systémů existuje celá řada, v následujících odstavcích je uvedeno pouze několik zástupců, především těch, které jsou zdarma popřípadě poskytují zkušební verze.

Oxygen Forensic® Suite

Forenzní software společnosti Oxygen Forensics[4], která kromě samotného softwaru poskytuje také školení forenzních specialistů a prodává hardware určený pro forenzní analýzu pomocí jejich nástroje. Aplikace Oxygen Forensic® Suite je k dispozici zdarma v omezené verzi[5], ve které jsou nedostupné některé pokročilejší[6] funkce.

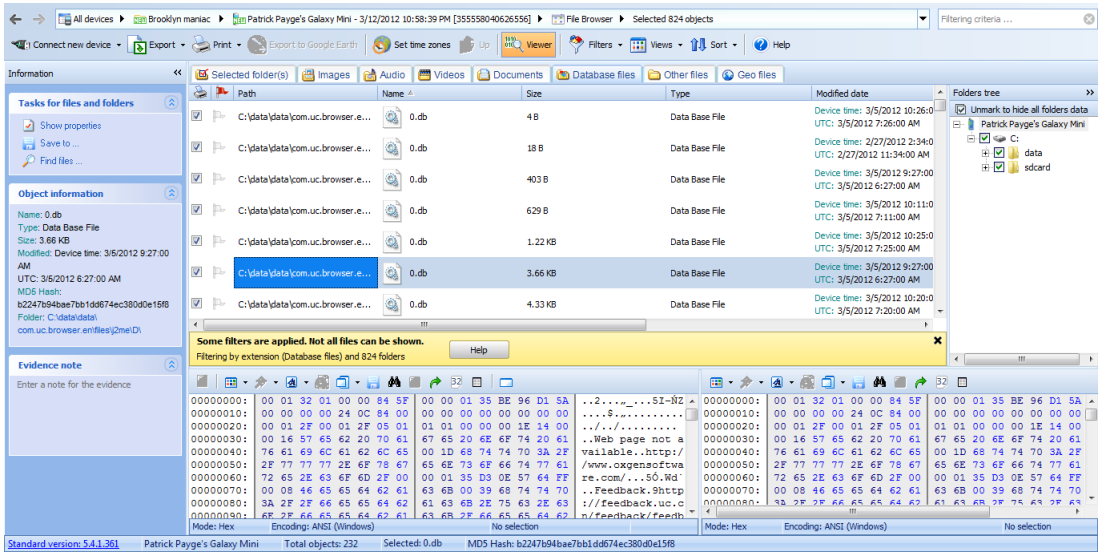
[3] Domovská stránka společnosti Cellebrite – http://www.cellebrite.com/
[4] Domovská stránka společnosti Oxygen Forensics – http://www.oxygen-forensic.com/en/
[5] Seznam funkcí verze *Standard* – http://www.oxygen-forensic.com/en/features/standard
[6] Seznam funkcí placené verze – http://www.oxygen-forensic.com/en/features/analyst

Oxygen Forensic® Suite se skládá ze dvou částí, první je tzv. Extractor, který zajišťuje připojení telefonu (popřípadě dříve vytvořeného obrazu) a získání dat. Na obrázku 2.2 je zobrazen průběh načítání dat z mnou připojeného telefonu.



Obrázek 2.2: Oxygen Forensic® Extractor – získávání dat z telefonu Samsung Galaxy Mini

Druhá část je samotný forenzní nástroj, který umožňuje prohlížení a analýzu dat a také vytvoření dokumentace nebo forenzního posudku. Na obrázku 2.3 vidíme zobrazení dat z demo obrazu zařízení dostupných ze stránek výrobce[7].



Obrázek 2.3: Oxygen Forensic® Suite – prohlížení dat z telefonu Samsung Galaxy Mini

[7]Možnost stažení vzorových dat – http://www.oxygen-forensic.com/en/download/devicebackups

MOBILedit! Forensic

MOBILedit! Forensic[8] je software české společnosti COMPELSON, které se forenzní analýze věnuje od roku 1996, kdy představili jiný úspěšný forenzní nástroj – SIMedit[9]. Opět je k dizpocici zdarma v omezené verzi.

Program umožňuje použít různé způsoby připojení mobilního zařízení, od kabelu až po wi-fi. Při prvním připojení analyzovaného zařízení s operačním systémem Android musí být aktivován režim *Ladění USB*, MOBILedit! Forensic totiž do zařízení instaluje malou aplikaci – ME! Forensic Connector. Ta zpřístupňuje výše zmiňované připojení přes wi-fi a dále zprostředkovává přístup k datům telefonu. Načtení dat z telefonu probíhá pomocí průvodce. Pokud z telefonu chceme získat i data o aplikacích, je nutné ručně pro každou aplikaci v telefonu potvrdit její zálohování do počítače.

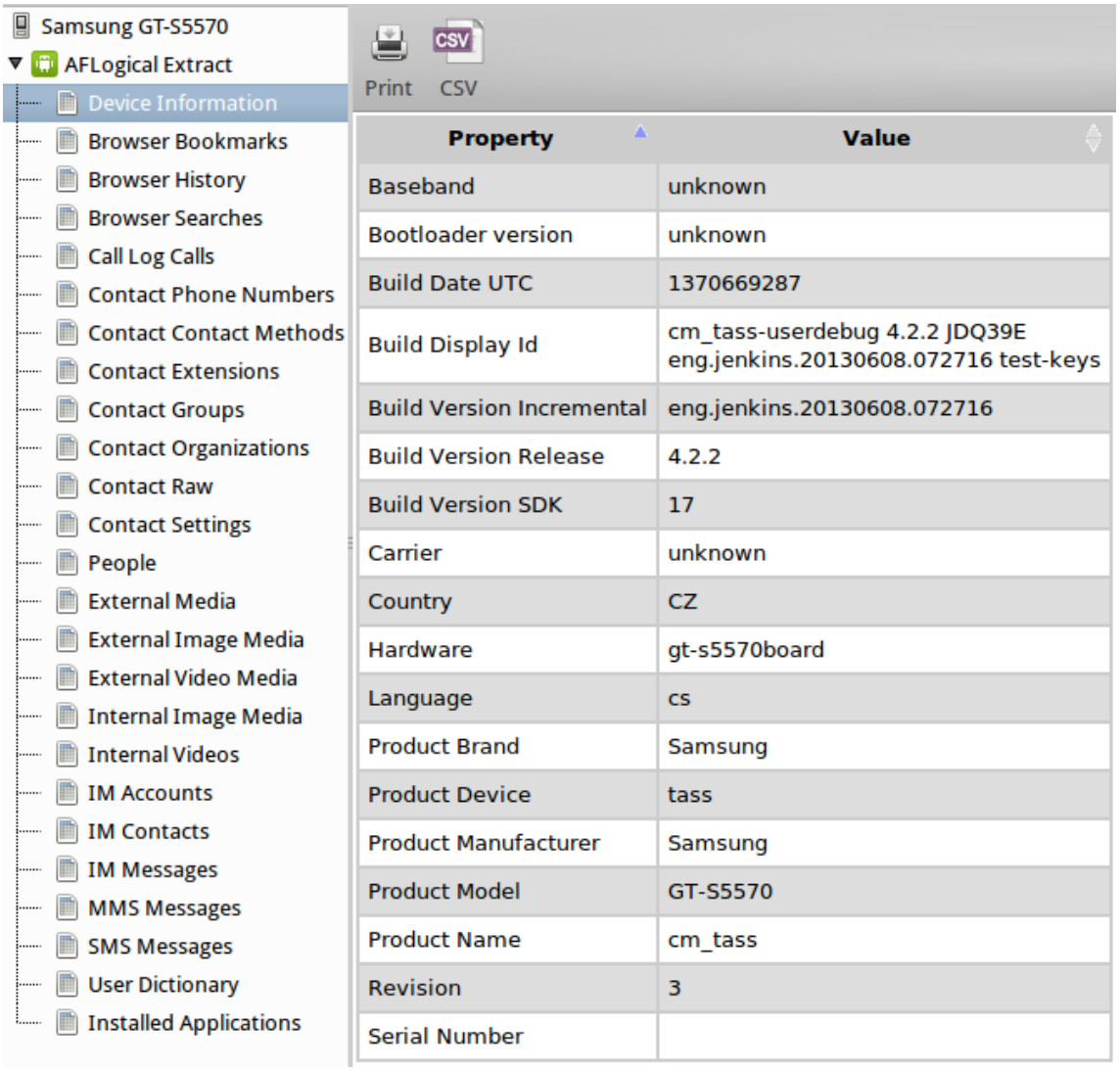
Obrázek 2.4: MOBILedit! Forensic Lite – Samsung Galaxy Mini

[8] Domovská stránka programu MobilEdit! – http://www.mobiledit.com/home.htm
[9] Historie společnosti COMPELSON – http://www.mobiledit.com/company

viaExtract

Jedná se o program společnosti viaForensics[10]. Je dodáván na virtuálním obrazu disku s přednastaveným Ubuntu[11] a nainstalovaným programem viaExtract. Program viaExtract mimo jiné obsahuje open source framework AFLogical[12], který umožňuje například odemčení připojeného telefonu, vytvoření obrazu paměti nebo vytvoření časové osy změn prováděných v telefonu.

Stejně jako výše popsaný MOBILedit! Forensic i tento nástroj potřebuje na zařízení aktivovaný režim *Ladění USB*, protože také instaluje aplikaci zajišťující spojení s počítačem. Data jsou následně získána pomocí jednoduchého průvodce.



Obrázek 2.5: viaExtract – analýza telefonu Samsung Galaxy Mini

[10] Domovská stránka programu viaExtract – https://viaforensics.com/products/viaextract/

[11] *The leading OS for PC, tablet, phone and cloud — Ubuntu* – http://www.ubuntu.com/

[12] Popis frameworku AFLogical – https://viaforensics.com/resources/tools/android-forensics-tool/

Santoku Linux

Santoku Linux[13] je linuxová distribuce, která obsahuje celou řadu (nejen forenzních) nástrojů (obrázek 2.6). Distribuce staví na Lubuntu[14] a obsahuje již nainstalované potřebné ovladače a SDK. Autoři doporučují instalaci do virtuálního počítače (VirtualBox[15] nebo VMware Player[16]).

Obrázek 2.6: Santoku Linux – nabídka programů

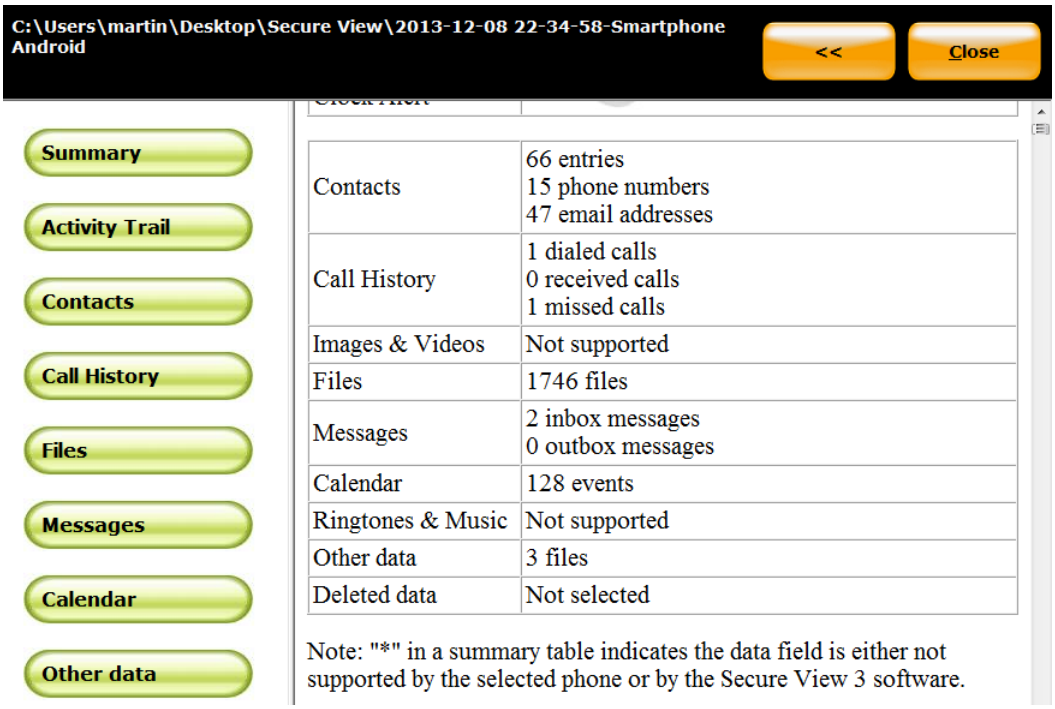
Secure View 3

Nástroj Secure View[17] americké společnosti Susteen[18] používá například i FBI[19]. Verze zdarma umožňuje pouze základní analýzu (obrázek 2.7), k dispozici jsou alespoň v programu uložené vzorové zprávy, které je možné prohlížet a udělat si tak obrázek o možnostech programu (obrázek 2.8).

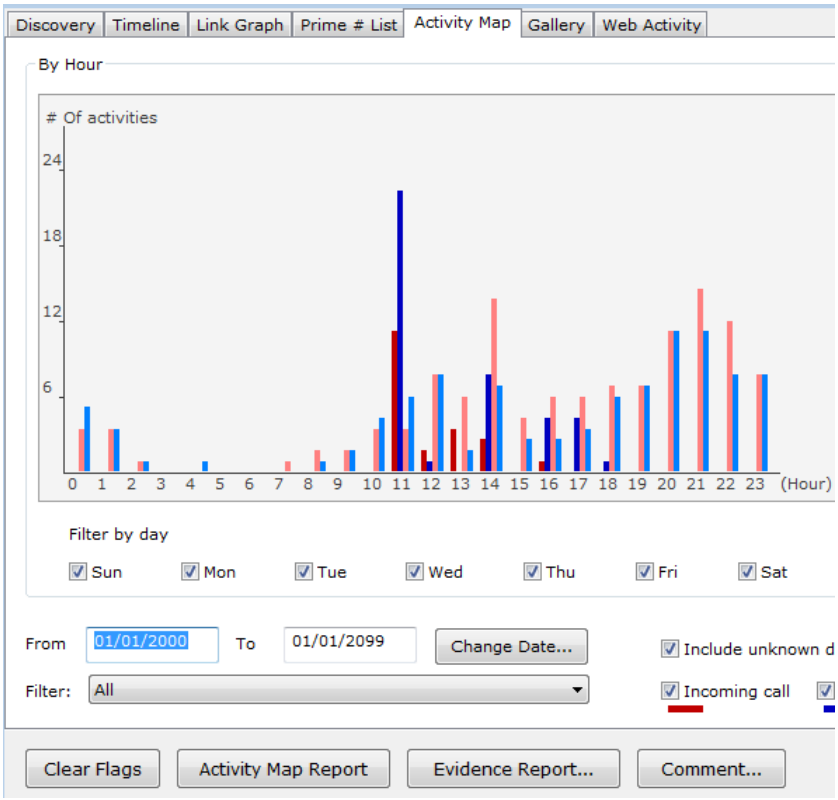
Další nástroje

Při výběru výše zmíněných nástrojů jsem narazil také na nástroje, které zde nejsou uvedené (SAFT[20], Mobile Forensic Examiner[21] a další). Do výběru jsem je nezařadil proto, že jejich funkcionalita nebyla tak rozsáhlá jako u výše zmíněných, nebo nedisponovaly zkušební verzí, případně měly další nedostatky.

[13] *Santoku-Linux* – https://santoku-linux.com/
[14] *lubuntu — lightweight, fast, easier* – http://lubuntu.net/
[15] *Oracle VM VirtualBox* – https://www.virtualbox.org/
[16] *VMware Player Plus: Easiest Way to Run a Virtual Machine* – http://www.vmware.com/cz/products/player/
[17] Informace o softwarovém kitu Secure View 3 – http://www.secureview.us/secureview3
[18] Více informací o spoečnosti Susteen – http://www.secureview.us/about
[19] Informace o objednáce softwaru pro FBI – https://www.fbo.gov/index?s=opportunity&mode=form&id=a544ee0cfae5804c0bde6b2de40755fa&tab=core&_cview=1
[20] *SAFT – Mobile Forensics* – http://www.signalsec.com/saft/
[21] *MPE+ Mobile Phone Forensics* – http://www.accessdata.com/products/digital-forensics/mobile-phone-examiner



Obrázek 2.7: Secure View 3 Trial – analýza telefonu Samsung Galaxy Mini



Obrázek 2.8: Secure View 3 Trial – prohlížení vzorových demo dat

Kapitola 3

OS Android

V této kapitole bude popsán operační systém Android[1], který je v současnosti největší mobilní platformou[2]. Je založen na linuxovém jádře (viz dále) a je optimalizován především pro dotyková zařízení. Na jeho vývoji se podílí tzv. Open Handset Alliance[3], která má za cíl vyvinout otevřený standard pro mobilní zařízení. OS Android byl poprvé veřejnosti představen na telefonu HTC (T-Mobile G1[4]) v roce 2008. [13]

Operační systém Android se neustále vyvíjí a proto existují zařízení s různými verzemi systému. V tabulce 3.1 je jejich stručný přehled aktuální k 2.12.2013[5].

Verze	**Název verze**	**Procento zařízení**
2.2	Froyo	1.6%
2.3.3 – 2.3.7	Gingerbread	24.1%
3.2	Honeycomb	0.1%
4.0.3 – 4.0.4	Ice Cream Sandwich	18.6%
4.1.x	Jelly Bean	37.4%
4.2.x		12.9%
4.3		4.2%
4.4	KitKat	1.1%

Tabulka 3.1: Přehled verzí OS Android

Operační systém Android je (kromě verze 3) prezentovaný jako open source[6], což umožňuje výrobcům zařízení upravovat systém tak, aby vyhovoval jejich hardwaru, nebo obsahoval jiné výchozí aplikace. Důsledkem této otevřenosti je také to, že tento systém můžeme nalézt na zařízeních s velice různorodou hardwarovou výbavou (například různé procesory, rozlišení obrazovky, velikost paměti).

[1] Domovské stránky operačního systému Android – http://www.android.com/

[2] *Android, the world's most popular mobile platform – Android Developers* – http://developer.android.com/about/index.html

[3] Domovské stránky projektu OHA (Open Handset Aliance) – http://www.openhandsetalliance.com/

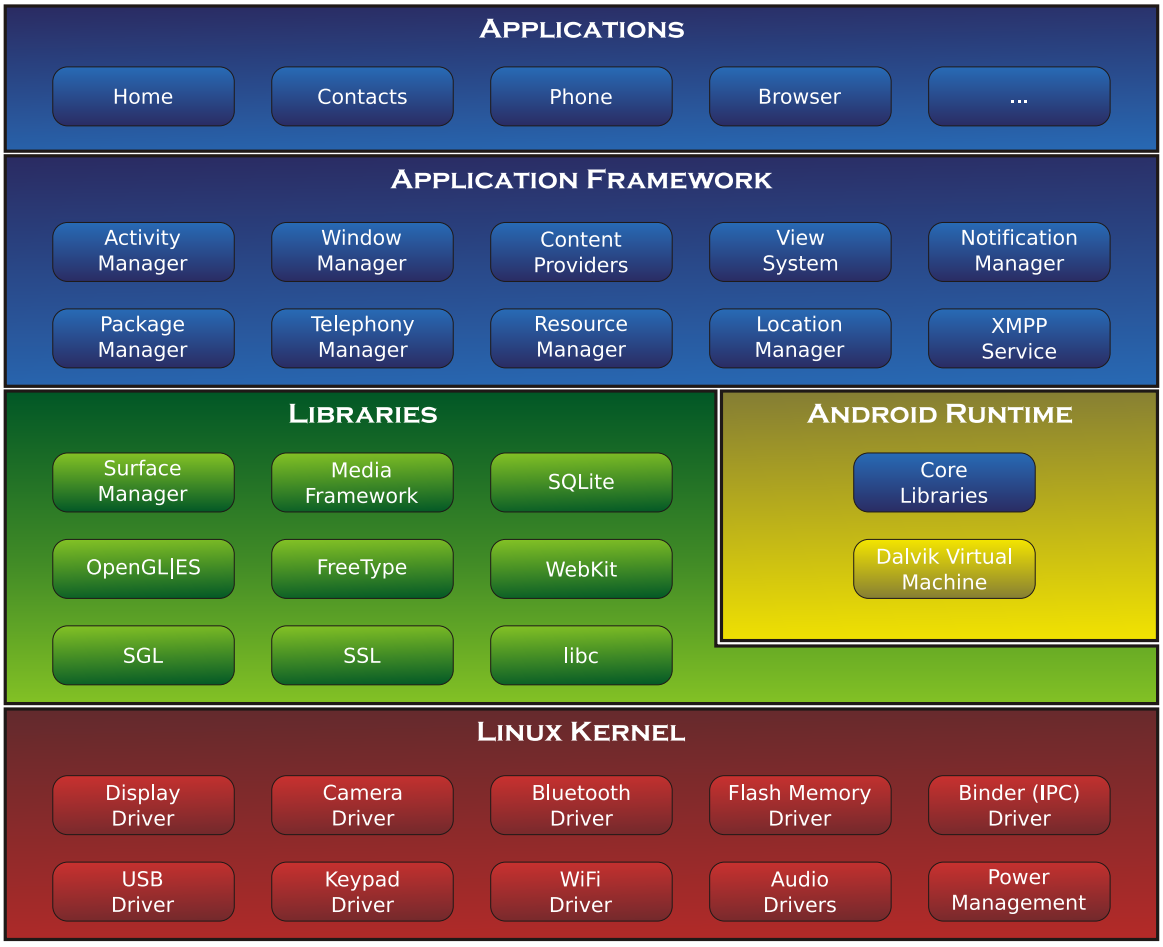
[4] Technické specifikace telefonu T-Mobile G1 – http://www.gsmarena.com/t_mobile_g1-2533.php

[5] Aktuální informace o podílu jednotlivých verzí na oficiálních stránkách pro vývojáře – http://developer.android.com/about/dashboards/index.html

[6] Stránky poskytující zdrojové kódy projektu Android Open Source – http://source.android.com/

3.1 Architektura

Tato podkapitola vychází z knihy [12]. Operační systém Android je tvořen několika vrstvami, z nichž každá má přesně definované chování a poskytuje určité služby vrstvě nad ní (obrázek 3.1). Nejspodnější vrstvou je linuxové jádro (Linux kernel). Nad tímto jádrem jsou knihovny (Libraries) a běhové prostředí (Android runtime). Další vrstvou je aplikační framework (Application framework), který dovoluje systému a aplikacím pracovat s knihovnami a jádrem. Poslední vrstvou jsou samotné aplikace (Applications) běžící v operačním systému Android.



Obrázek 3.1: Vrstvy OS Android. Zdroj: [5]

Linuxové jádro

Linuxové jádro v OS Android je upraveno pro běh na mobilních zařízeních, je proto ochuzeno o některé funkce tradiční linuxové distribuce (například chybí některé GNU utility). Výhodou použití linuxového jádra je abstrakce hardwaru, proto je možné operační systém Android spustit na velkém množství různých zařízení. Dále je využito základních funkcí operačního systému – správa procesů, paměti a vstupně-výstupních operací. Různé verze OS Android pracují s různými verzemi linuxového jádra (viz tabulka 3.2).

Verze Androidu	Verze linuxového jádra
Android 2.3.x	Linux Kernel 2.6.35
Android 4.x	Linux Kernel 3.0.1

Tabulka 3.2: Přehled verzí linuxových jader v různých verzích OS Android

Knihovny

OS Android obsahuje řadu C/C++ knihoven, které jsou používány jednotlivými součástmi operačního systému. Vývojářům aplikací jsou tyto knihovny dostupné přes aplikační framework (viz dále), samotné aplikace (napsány v Javě) je používají pomocí JNI (Java Native Interface).

Běhové prostředí

Běhové prostředí se skládá ze dvou částí: Android Core Libraries a Dalvik VM.

Knihovny Android Core Libraries poskytují funkcionalitu knihoven programovacího rozhraní Javy (API), ale jsou částečně modifikovány – neobsahují například AWT nebo Swing.

Virtuální stroj Dalvik VM je modifikace Java Virtual Machine (JVM), při spouštění aplikace přidává jeden krok navíc – rekompiluje soubory `.class` do formátu `.dex`[7]. Výhodou použití virtuálního stroje je, že spouštěné aplikace jsou sandboxované a tak se nemohou vzájemně přímo ovlivňovat a také nemají přístup do systému nebo k hardwaru.

V nejnovější verzi Android 4.4 je pro testování dostupné nové běhové prostředí – ART[8], které má do budoucna nahradit Dalvik VM.

Aplikační framework

Aplikační framework poskytuje vývojářům celou řadu služeb. V tabulce 3.3 je uveden přehled hlavních tříd, které aplikační framework poskytuje.

Služba	Popis
Activity Manager	Spravuje životní cyklus aplikace.
Resource Manager	Umožňuje přístup ke zdrojům (např. texty, grafika...).
Telephony Manager	Poskytuje informace o telefoních službách v zařízení.
Location Manager	Poskytuje podporu pro získání polohy (např. z GPS).
Notification Manager	Umožňuje aplikacím získávat upozornění o některých událostech (např. o příchozím emailu).
Package Manager	Je zodpovědný za instalování aplikací a za informace o nainstalovaných aplikacích.
Content Providers	Umožňuje aplikacím přistupovat k datům z jiných aplikací, nebo sdílet svoje data s dalšími aplikacemi.
Views	Poskytuje různé možnosti zobrazení.

Tabulka 3.3: Přehled služeb aplikačního frameworku

[7] Popis formátu `.dex` – http://source.android.com/devices/tech/dalvik/dex-format.html
[8] Informace o běhovém prostředí ART – http://source.android.com/devices/tech/dalvik/art.html

Aplikace

Aplikace pro OS Android jsou psány v programovacím jazyku Java. Výrobci zařízení poskytují systém s kolekcí výchozích programů – například prohlížeč, kalendář, e-mailový klient, přehrávač multimédií... Další aplikace je možné získat z nejrůznějších zdrojů, nejznámější a největší je obchod s aplikacemi Google Play[9].

3.2 Uložení dat

V této části se bude popsáno jak operační systém Android pracuje s daty. Popíši jednotlivé druhy úložišť a následně souborové systémy, které jsou na těchto úložištích použity. Následující odstavce vycházejí z [9].

3.2.1 Druhy pamětí

Zařízení s OS Android využívají několik druhů fyzických úložišť dat. Samotný systém a data aplikací jsou uložena ve vnitřní paměti zařízení, dočasné soubory a data, se kterými systém pracuje, jsou uložena v paměti RAM. Uživatelská data a některá data aplikací systém ukládá na paměťovou kartu, fyzickou nebo virtuální. Existují také zařízení, které mají pouze vnitřní paměť, a nemají slot pro paměťové karty.

Vnitřní paměť

Vnitřní paměť bývá tvořena NAND pamětí[10], která je připájena na základní desku telefonu. Fakt, že je paměť takto napevno připevněná, znemožňuje její vyjmutí a následnou forenzní analýzu mimo zařízení. Do této paměti jsou data ukládána s přísnými bezpečnostními pravidly – aplikace nemají přístup k jiným než svým složkám (pokud neuvažujeme *root* zařízení, viz kapitola 3.3) a je zde pevně stanovená adresářová struktura.

Do paměti typu NAND není možné přistupovat náhodně, její kapacita je rozdělena alokační jednotky – stránky. Tyto stránky bývají veliké 512B nebo 2kB. Stránky jsou organizovány do větších celků – bloků. NAND paměť umožňuje provádět operace čtení a zápis po jednotlivých stránkách, operace mazání je však možná provádět pouze nad celými bloky.

Adresářová struktura konkrétního zařízení je zobrazena ve výpisu 3.1.

```
-rw-r--r-- root    root    13444 1970-01-01 01:00 TASS.rle
drwxr-xr-x root    root          2014-04-13 15:22 acct
drwxrwx--x system  cache         2014-05-02 09:48 cache
dr-x------ root    root          2014-04-13 15:22 config
lrwxrwxrwx root    root          2014-04-13 15:22 d -> /sys/kernel/debug
drwxrwx--x system  system        2013-12-26 19:14 data
-rw-r--r-- root    root      116 1970-01-01 01:00 default.prop
drwxr-xr-x root    root          2014-04-13 15:26 dev
lrwxrwxrwx root    root          2014-04-13 15:22 etc -> /system/etc
-rwxr-x--- root    root   117708 1970-01-01 01:00 init
-rwxr-x--- root    root     1328 1970-01-01 01:00 init.cm.rc
-rwxr-x--- root    root     2770 1970-01-01 01:00 init.goldfish.rc
```

[9] *Google Play* – https://play.google.com/store

[10] Dokument věnující se technologiím NAND pamětí – http://www.mikrozone.sk/soubory/downloads/print/dps-az/2/soucastky-pohled_na_nand.pdf

```
-rwxr-x--- root     root      2264 1970-01-01 01:00 init.gt-s5570board.bluetooth.rc
-rwxr-x--- root     root       393 1970-01-01 01:00 init.gt-s5570board.parts.rc
-rwxr-x--- root     root      4652 1970-01-01 01:00 init.gt-s5570board.rc
-rwxr-x--- root     root      5166 1970-01-01 01:00 init.gt-s5570board.usb.rc
-rwxr-x--- root     root     21951 1970-01-01 01:00 init.rc
-rwxr-x--- root     root       528 1970-01-01 01:00 init.recovery.gt-s5570board.rc
-rwxr-x--- root     root       301 1970-01-01 01:00 init.superuser.rc
-rwxr-x--- root     root      1795 1970-01-01 01:00 init.trace.rc
-rwxr-x--- root     root      3947 1970-01-01 01:00 init.usb.rc
drwxr-xr-x root     root           1970-01-01 01:00 lib
drwxrwxr-x root     system         2014-04-13 15:22 mnt
dr-xr-xr-x root     root           1970-01-01 01:00 proc
drwx------ root     root           2013-10-08 21:16 root
drwxr-x--- root     root           1970-01-01 01:00 sbin
drwxrwx--x system   system         2014-04-13 15:22 sd-ext
lrwxrwxrwx root     root           2014-04-13 15:22 sdcard -> /storage/sdcard0
d---r-x--- root     sdcard_r       2014-04-13 15:22 storage
drwxr-xr-x root     root           2014-04-13 15:22 sys
drwxr-xr-x root     root           2013-12-26 15:37 system
-rw-r--r-- root     root       272 1970-01-01 01:00 ueventd.goldfish.rc
-rw-r--r-- root     root      3755 1970-01-01 01:00 ueventd.gt-s5570board.rc
-rw-r--r-- root     root      5897 1970-01-01 01:00 ueventd.rc
lrwxrwxrwx root     root           2014-04-13 15:22 vendor -> /system/vendor
```

Výpis kódu 3.1: Výpis příkazu `ls -l`

RAM

Paměť RAM funguje v podstatě stejně jako v počítačích. Slouží k manipulaci s částmi operačního systému, aplikací nebo dat. Jedná se o paměť typu *volatile*, takže její obsah není při restartování zařízení uložen. To komplikuje její využití pro forenzní analýzu. Je potřeba vytvořit její obraz aniž by došlo k vypnutí zařízení. Tato paměť může obsahovat užitečná data týkající se spuštěných aplikací – například přihlašovací jména, hesla či cookies.

Podle [7] k vytvoření obrazu paměti dané aplikace obsahuje operační systém Android přímo nástroje – stačí zaslat procesu signál `SIGUSR1`. Obraz celé paměti je možný vytvořit pouze s právy *roota*.

Paměťová karta

Paměťová karta má na rozdíl od vnitřní paměti mírnější bezpečností politiky. Je to dáno tím, že se musí počítat s vyjmutím paměťové karty a jejím následným vložením do jiného zařízení. Tímto zařízením nemusí být jen další mobilní zařízení s OS Android, ale například počítač s různým operačním systémem.

Možnost vyjmout paměťovou kartu usnadňuje její forenzní analýzu. Lze vytvořit bitovou kopii karty na jiném zařízení a tuto kopii poté zkoumat.

3.2.2 Souborové systémy

Jelikož je Android postavený na linuxovém kernelu (viz předchozí kapitoly), podporuje celou řadu souborových systémů. Jejich seznam lze zjistit z obsahu souboru `/proc/filesytem`. Níže uvádím obsah tohoto souboru ze dvou zařízení: 3.2 – tablet[11] a 3.3 – mobilní telefon[12].

```
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   cgroup
nodev   tmpfs
nodev   devtmpfs
nodev   binfmt_misc
nodev   debugfs
nodev   securityfs
nodev   sockfs
nodev   usbfs
nodev   pipefs
nodev   anon_inodefs
nodev   devpts
        ext3
        ext2
        ext4
nodev   ramfs
        vfat
        msdos
        iso9660
        hfsplus
        ntfs
nodev   fuse
        fuseblk
nodev   fusectl
        udf
        yaffs
        yaffs2
nodev   mtd_inodefs
nodev   ubifs
```

Výpis kódu 3.2: Tablet

```
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   cgroup
nodev   tmpfs
nodev   debugfs
nodev   sockfs
nodev   pipefs
nodev   anon_inodefs
nodev   rpc_pipefs
nodev   devpts
        ext3
        ext2
        ext4
nodev   ramfs
        vfat
nodev   fuse
        fuseblk
nodev   nfs
```

Výpis kódu 3.3: Mobilní telefon

Pro ukládání uživatelských dat se v operačním systému Android používají pouze některé z podporovaných:

YAFFS2

Yaffs (Yet Another Flash File System) je souborový systém speciálně vyvinut pro flash paměti. YAFFS2 používaný v zařízeních s OS Android je přizpůsobený pro NAND paměti

[11] Ainol Novo 7 Crystal, Android 4.1.1
[12] Samsung Galaxy Mini, Android 4.2.2 (CyanogenMod 10.1.6)

s 2kB velkými stránkami.

Ext4

S vydáním Androidu 2.3 Gingerbread došlo k opuštění souborového systému YAFSS2. Nově začal být používán souborový systém Ext4, který je standardním souborovým systémem v operačním systému Linux. Důvodem k přechodu bylo větší zabezpečení při ukládání dat[13].

FAT32

FAT32 se v OS Android používá především jako souborový systém paměťové karty. Důvod pro využití tohoto souborového systému je podpora napříč platformami. Předpokládá se, že paměťová karta nebude využívána pouze v daném zařízení. Také je tím umožněno připojit zařízení k počítači a zpřístupnit úložiště (paměťovou kartu) jako *USB mass storage*[14]. Kvůli možnosti přesunout a spouštět aplikace z SD karty bylo nutné zajistit příslušná přístupová práva i při spouštění z FAT32, aplikace je proto zašifrována klíčem, který je specifický pro konkrétní HW, to znemožňuje její spuštění na jiném zařízení.[15]

3.3 Práva uživatelů

V OS Android existují dva druhy přístupových práv uživatelů. Podobně jako v Linuxu, existují takzvaní běžní uživatelé s omezenými právy a speciální uživatel – *root*, ten má neomezený přístup k celému systému. V zařízeních s OS Android bývá od výroby běžně nastaven režim s omezeným přístupem. Od Androidu verze 4.2[16] je navíc k dispozici využití více uživatelských účtů na jednom zařízení. Pro jednotlivé uživatele systém ukládá data separátně. Práva jsou nastavena tak, aby běžného uživatele neomezovaly, může instalovat aplikace, pracovat se svými soubory atd. Nemůže však zasahovat hlouběji do systému a například odinstalovat aplikace od výrobce zařízení. [3]

Root

Existují však nástroje, které umožňují odemknout uživatele *root* a získat tak přístup k celému souborovému systému a k dalším datům, která jsou pro běžného uživatele skryta. Postupů provedení takzvaného *rootu* zařízení existuje celá řada. Některé můžeme nalézt v knize [14], popřípadě v diskusních fórech na internetu[17]. Například pro tablet Ainol Novo 7 Crystal stačí stáhnout potřebný instalační balíček, ten uložit na SD kartu a z takzvaného Recovery Modu tento balíček nainstalovat[18].

[13] Příspěvek na blogu vývojářů OS Android, který se týká bezpečnosti ukládání dat – http://android-developers.blogspot.cz/2010/12/saving-data-safely.html

[14] *Universal Serial Bus Mass Storage Class Specification Overview* – http://www.usb.org/developers/devclass_docs/usb_msc_overview_1.2.pdf

[15] Popis spouštění aplikací z SD karty – http://android-developers.blogspot.cz/2010/07/apps-on-sd-card-details.html

[16] Popis API pro víceuživatelské prostředí v operačním systému Android – http://developer.android.com/about/versions/android-4.2.html#MultipleUsers

[17] Například *xda-developers* – http://forum.xda-developers.com/index.php

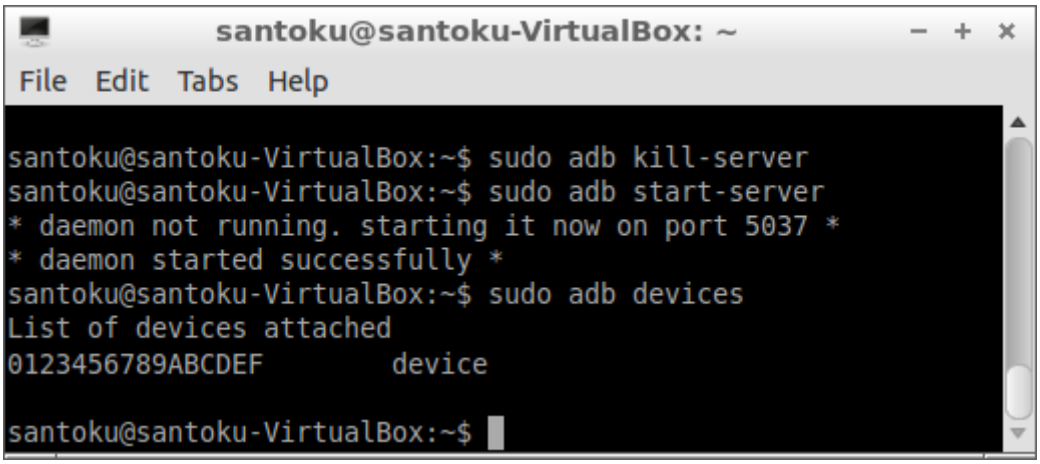
[18] *Návod jak provést root tabletu Ainol Novo 7 CRYSTAL* – http://androidforum.cz/root-tabletu-ainol-novo-7-crystal-t33358.html

Dále existují přímo alternativní *ROM*[19], které mají uživatele *root* povoleny ihned po nainstalování (například známý CyanogenMod[20]).

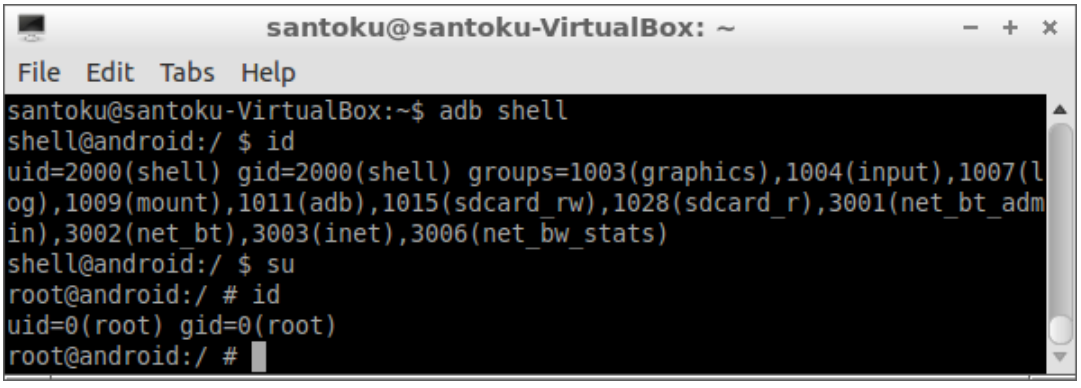
Pro forenzní analýzu vnitřní paměti zařízení jsou výhodnější práva uživatele *root*, která umožní získat obraz celé paměti.

3.4 ADB

Podle [1] je Android Debug Bridge (zkráceně ADB) univerzální nástroj pracující přes příkazovou řádku, který umožňuje komunikaci se zařízením připojeným k počítači. Pracuje na principu klient-server. Server běží na portu 5037 a zprostředkovává komunikaci mezi klientem (v počítači) a připojeným zařízením. Zařízení, které chceme ovládat pomocí ADB musí být připojeno v režimu *Ladění USB*. Tento režim musíme na zařízení zapnout (`Nastavení -> Systém -> Pro vývojáře -> Ladění -> Ladění USB`).



Obrázek 3.2: ADB – spuštění serveru a výpis připojených zařízení



Obrázek 3.3: ADB – identifikace běžného uživatele a roota

[19] Různé oficiální i neoficiální verze firmwaru zařízení.

[20] *CyanogenMod – Android Community Operating System* – http://www.cyanogenmod.org/

3.5 Rizika

Aplikace pro operační systém Android často spoléhají zabezpečením pouze na systém práv v systému. Pokud si uživatel, nebo někdo, komu se zařízení dostane do rukou, provede *root*, jsou data aplikací čitelná bez větších problémů. Většina dat je totiž ukládána v čistém plaintextu (například přihlašovací jména a hesla v prohlížeči). Jakákoliv aplikace, která dostane povolení *root*, tak tato data může bez problému číst, popřípadě je kdokoliv může stáhnout do počítače pomocí ADB.

Andrew Hoog uvádí ve své prezentaci [10] příklad pro konkrétní aplikaci Any.DO[21]. Při implementaci aplikace DPForensic (viz dále) jsem se mohl přesvědčit, že tato aplikace není jediná, například samotný operační systém ukládá přihlašovací údaje k přístupovým bodům bezdrátových sítí s souboru v čisté textové podobě (ve výpisu 3.4 je ukázka konfiguračního souboru z mobilního telefonu Samsung Galaxy Mini).

```
ctrl_interface=wlan0
update_config=1
device_name=cm_tass
manufacturer=Samsung
model_name=GT-S5570
model_number=GT-S5570
serial_number=
device_type=10-0050F204-5
config_methods=physical_display virtual_push_button keypad

network={
        ssid="Internet"
        key_mgmt=NONE
        priority=1
}

network={
        ssid="SoftAP"
        psk="Heslo123Heslo123"
        key_mgmt=WPA-PSK
        priority=3
}
```

Výpis kódu 3.4: Část souboru `wpa_supplicant.conf`

[21] *Any.DO* – http://www.any.do/

Kapitola 4

Specifikace

Cílem je vytvořit aplikaci, která umožní analyzovat data v zařízení s operačním systémem Android a na základě těchto dat odhalit slabá místa popř. získat užitečné informace.

Většina dostupných forenzních nástrojů je určena pro práci v laboratoři, kde je daný software spuštěný na klasickém počítači s operačním systémem Windows nebo Linux (viz kapitola 2.1.1). Jediná aplikace, která je určena pro forenzní analýzu přímo z daného zařízení s OS Android je AFLogical Open Source Edition[1]. Tato aplikace dokáže z přístroje získat MMS, SMS, kontakty a výpis volání. Získaná data ukládá ve formátu `CSV` na paměťovou kartu vloženou do zařízení.

Jelikož i výše zmíněné forenzní nástroje do analyzovaného zařízení instalují aplikaci, nemělo by instalovaní a spouštění vytvořené aplikace přímo do zařízení výrazně ovlivňovat zkoumaný přístroj.

Aplikace bude určena pro OS Android (verze 4.0 a vyšší) a bude disponovat následujícími vlastnostmi:

- Přístup k sms, adresáři kontaktů, kalendáři…,
- přístup k celému souborovému systému (root),
- analýza dat nainstalovaných aplikací,
- analýza historie prohlížeče,
- mazání citlivých dat.

Aplikace bude analyzovat pouze data z vnitřní paměti přístroje, popřípadě data aplikací nainstalovaných na paměťové kartě. Analýze samotných souborů na paměťové kartě (fotky, hudební soubory, smazaná data…) se věnovat nebude, protože paměťovou kartu je snadné vyjmout a analyzovat speciálními nástroji.

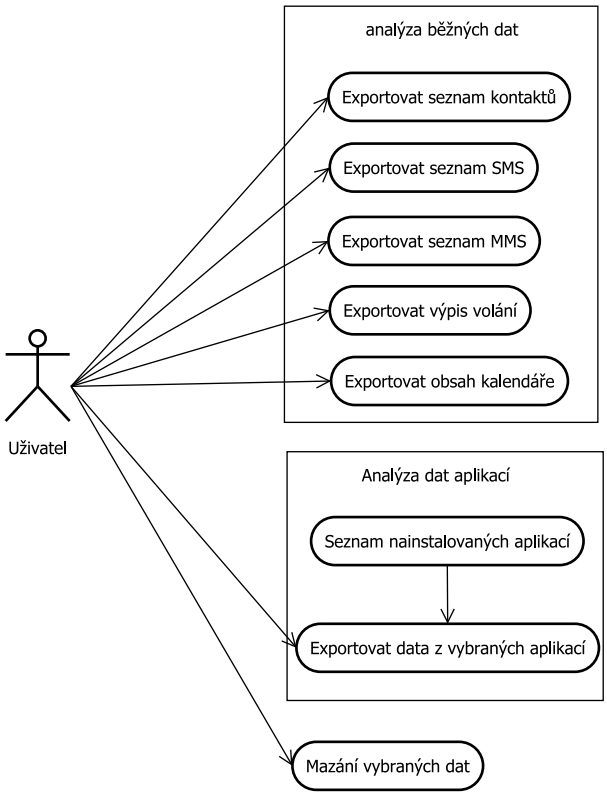
[1] *AFLogical OSE – viaForensics* – https://viaforensics.com/resources/tools/android-forensics-tool/

Kapitola 5

Návrh

Jelikož se jedná o aplikaci pro operační systém Android, bude napsána v jazyce Java a používat Android API[1]

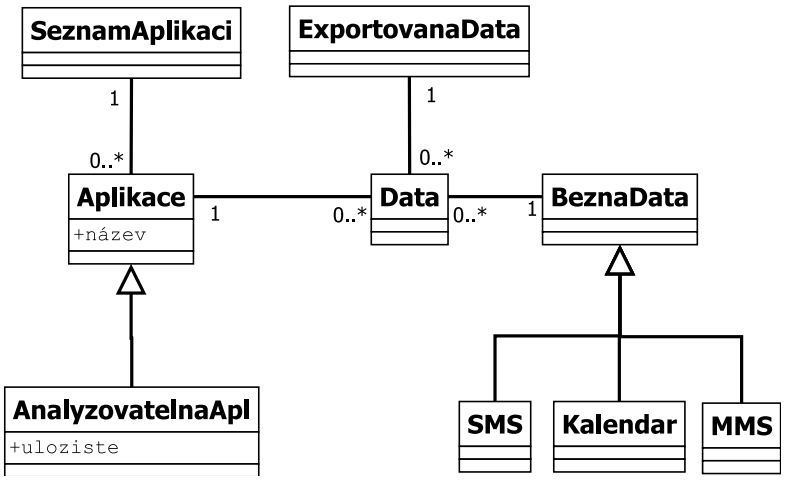
Na obrázku 5.1 je uveden diagram případu užití. Uživatel bude moci vybrat, která data mají být získána. Běžně dostupná data přístroje budou získána rovnou, u dat aplikací bude nejdříve zobrazen seznam nainstalovaných aplikací a u podporovaných aplikací bude nabídnuto získání dat. Pokud to konkrétní situace umožní, bude také možné vybraná data smazat.



Obrázek 5.1: Diagram užití

[1] Aktuální verze Android API – http://developer.android.com/about/versions/android-4.4.html

Obrázek 5.2 zobrazuje část diagramu tříd, která se týká získávaných dat. Data, v diagramu označená jako `BeznaData`, jsou ze zařízení získána přes poskytované API. Data aplikací je nutné vyčíst přímo z úložiště konkrétní aplikace. Přístup k tomuto úložišti není běžně k dispozici (přistupovat k němu smí pouze aplikace, které toto uložiště náleží). Pokud je však v zařízení proveden *root* (viz kapitola 3.3), je toto úložiště dostupné i cizím aplikacím.



Obrázek 5.2: Diagram tříd

5.1 Data aplikací

Aplikace se zaměří převážně na data z aplikací, které v operačním systému Android aktivně používám na svých zařízeních. Je pravděpodobné, že tyto aplikace obsahují osobní data vhodná k analýze. Jedná se o aplikace pro sociiální sítě (Facebook[2], Twitter[3], Foursquare[4]), komunikační programy (Google Hangouts[5], Facebook Messenger[6]), mapové aplikace (Mapy.cz[7], Google mapy[8]), poznámky (Google Keep[9]) a webový prohlížeč.

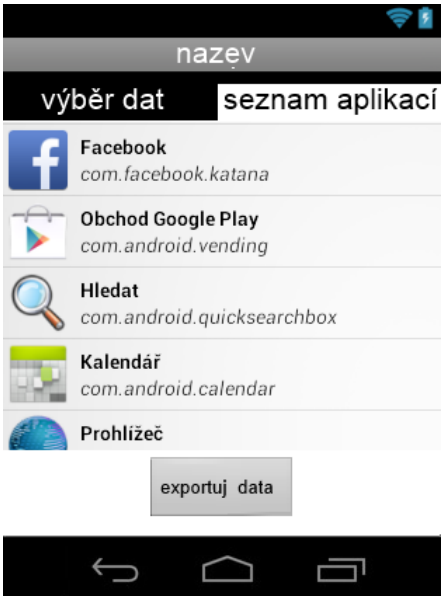
Pro případné rozšíření o další aplikace bude při implementaci snaha o co nejjednodušší možnost přidaní.

[2] *Facebook* – https://www.facebook.com/
[3] *Twitter* – https://twitter.com/
[4] *Foursquare* – https://foursquare.com/
[5] *Google+ Hangouty* – http://www.google.com/+/learnmore/hangouts/?hl=cs
[6] *Messenger* – http://cs-cz.facebook.com/mobile/messenger
[7] *Mapy.cz* – http://mapy.cz/
[8] *Google Maps* – https://maps.google.com/
[9] *Google Keep* – https://keep.google.com/

5.2 Uživatelské rozhraní

Návrh uživatelského rozhraní bude zaměřen na jednoduchost použití celé aplikace. Výběr dat a export bude probíhat na jedné obrazovce, další obrazovka zobrazí seznam nainstalovaných aplikací se zvýrazněním podporovaných aplikací.

Obrázek 5.3: Návrh uživatelského rozhraní – běžná data



Obrázek 5.4: Návrh uživatelského rozhraní – data z aplikací

Kapitola 6

Implementace aplikace

V této kapitole popíši umístění dat jednotlivých aplikací a implementaci navržené aplikace pro OS Android, její součásti a uživatelské rozhraní a také způsob ovládání vytvořené aplikace. Na konci kapitoly zmíním postupy optimalizace pro mobilní zařízení, požadavky implementované aplikace a krátce se věnuji testování.

K vývoji aplikace bylo využito vývojové prostředí Eclipse[1] se zásuvným modulem ADT[2], který obsahuje Android SDK[3]. V průběhu vývoje byla aplikace testována na skutečných zařízeních (mobilní telefon[4] a tablet[5]) kvůli přítomnosti reálných dat, která by v emulátoru (součást SDK) bylo nutné nejprve vytvořit. Do zařízení se aplikace nahrává jako soubor s příponou `.apk`, ze kterého je aplikace nainstalována.

Vytvořená aplikace se jmenuje `DPForensic`, název balíku je `cz.vutbr.fit.dpforensic`. Zdrojové kódy se nachází ve složce `DPforensic\src`.

6.1 Data aplikací

Data z mobilního zařízení jsou získávána dvojím způsobem – využitím poskytovaného API a přímým čtením z SQLite databází[6].

Pomocí veřejného API je možné získat především kontakty a události kalendáře. API pro práci s kontakty je v OS Android od začátku[7], v páté verzi API se ale přešlo na nový způsob[8]. První metodu již není doporučeno používat. Aplikace využívající API pro čtení kontaktů musí mít uvedeno oprávnění `READ_CONTACTS`[9]. API pro čtení kalendáře bylo přidáno až ve verzi 14[10]. Aplikace využívající toto API musí uvádět oprávnění `READ_CALENDAR`[11].

[1] *Eclipse – The Eclipse Foundation open source community website* – http://www.eclipse.org/
[2] *ADT Plugin* – http://developer.android.com/tools/sdk/eclipse-adt.html
[3] *Exploring the SDK* – http://developer.android.com/sdk/exploring.html
[4] Samsung Galaxy Mini, Android 4.2.2 (CyanogenMod 10.1.6)
[5] Ainol Novo 7 Crystal, Android 4.1.1
[6] *Saving Data in SQL Databases* – http://developer.android.com/training/basics/data-storage/databases.html
[7] *Contacts* – http://developer.android.com/reference/android/provider/Contacts.html
[8] *ContactsContract* – http://developer.android.com/reference/android/provider/ContactsContract.html
[9] *Manifest.permission* – http://developer.android.com/reference/android/Manifest.permission.html#READ_CONTACTS
[10] *CalendarContract* – http://developer.android.com/reference/android/provider/CalendarContract.html
[11] *Manifest.permission* – http://developer.android.com/reference/android/Manifest.permission.html#READ_CALENDAR

Pro získání SMS zpráv slouží `Uri`[12] `content://sms`. Aplikace, která přistupuje k této Uri, musí mít uvedeno oprávnění `READ_SMS`[13].

Nainstalované aplikace ukládají data v SQLite databázích ve své složce – `/data/data/název_aplikace/databases`. V této složce se většinou nachází několik `.db` nebo `.db2` souborů s databázemi. Jelikož bylo potřeba vytvořit analyzátor pro jednotlivé aplikace zvlášť, byly databáze nejdříve prozkoumány na počítači pomocí programu **SQLite Database Browser**[14] a doplňku pro Mozilla Firefox – **SQLite Manager**[15]. V následujících odstavcích popíši strukturu databáze podporovaných aplikací. Podporu pro další aplikace je možné přidat v budoucnosti rozšířením stávajícího analyzátoru.

Prohlížeč

Výchozí internetový prohlížeč v OS Android má název `com.android.browser`. Ve složce s databázemi jsou nejzajímavější tři soubory: `autofill.db`, `browser2.db` a `webview.db`.

Databáze `autofill` (výpis 6.1) obsahuje tabulku, ve které jsou uloženy data automatického dokončování. Můžeme tak odtud získat informace o tom, co uživatel vyplňuje ve webových formulářích. V databázi `webview` (výpis 6.2) je zajímavých tabulek hned několik. Jedná se o tabulky obsahující uložené přihlašovací údaje. Uživatelská jména a hesla jsou v tabulkách uloženy v čitelné podobě. Databáze `browser2` (výpis 6.3) obsahuje tabulky ukládající historii prohlížení a vyhledávání, uložené záložky a také obrázky (náhledy a ikony) navštívených stránek.

```
CREATE TABLE profiles (_id INTEGER PRIMARY KEY,fullname TEXT,email TEXT,
    companyname TEXT,addressline1 TEXT,addressline2 TEXT,city TEXT,state
    TEXT,zipcode TEXT,country TEXT,phone TEXT );
```

Výpis kódu 6.1: Struktura databáze `autofill.db`

```
CREATE TABLE formdata (_id INTEGER PRIMARY KEY, urlid INTEGER, name TEXT,
    value TEXT, UNIQUE (urlid, name, value) ON CONFLICT IGNORE);
CREATE TABLE formurl (_id INTEGER PRIMARY KEY, url TEXT);
CREATE TABLE httpauth (_id INTEGER PRIMARY KEY, host TEXT, realm TEXT,
    username TEXT, password TEXT, UNIQUE (host,realm) ON CONFLICT REPLACE);
CREATE TABLE password (_id INTEGER PRIMARY KEY, host TEXT, username TEXT,
    password TEXT, UNIQUE (host, username) ON CONFLICT REPLACE);
```

Výpis kódu 6.2: Struktura databáze `webview.db`

```
CREATE TABLE bookmarks(_id INTEGER PRIMARY KEY AUTOINCREMENT,title TEXT,url
     TEXT,folder INTEGER NOT NULL DEFAULT 0,parent INTEGER,position INTEGER
     NOT NULL,insert_after INTEGER,deleted INTEGER NOT NULL DEFAULT 0,
    account_name TEXT,account_type TEXT,sourceid TEXT,version INTEGER NOT
    NULL DEFAULT 1,created INTEGER,modified INTEGER,dirty INTEGER NOT NULL
    DEFAULT 0,sync1 TEXT,sync2 TEXT,sync3 TEXT,sync4 TEXT,sync5 TEXT);
```

[12] *Uri* – http://developer.android.com/reference/android/net/Uri.html

[13] *Manifest.permission* – http://developer.android.com/reference/android/Manifest.permission.html#READ_SMS

[14] *SQLite Database Browser* – http://sqlitebrowser.sourceforge.net/

[15] *SQLite Manager :: Doplňky aplikace Firefox* – https://addons.mozilla.org/cs/firefox/addon/sqlite-manager/

```
CREATE TABLE history(_id INTEGER PRIMARY KEY AUTOINCREMENT,title TEXT,url
    TEXT NOT NULL,created INTEGER,date INTEGER,visits INTEGER NOT NULL
    DEFAULT 0,user_entered INTEGER);
CREATE TABLE images (url_key TEXT UNIQUE NOT NULL,favicon BLOB,thumbnail
    BLOB,touch_icon BLOB);
CREATE TABLE searches (_id INTEGER PRIMARY KEY AUTOINCREMENT,search TEXT,
    date LONG);
CREATE TABLE settings (key TEXT PRIMARY KEY,value TEXT NOT NULL);
CREATE TABLE thumbnails (_id INTEGER PRIMARY KEY,thumbnail BLOB NOT NULL);
```

Výpis kódu 6.3: Struktura databáze `browser2.db`

Hangouts

Komunikační program od Google[16], dříve známý pod názvem Google Talk. Celý název balíku zní `com.google.android.talk`. Data ukládá do dvou databází `babel0.db` a `babel1.db`. Tyto databáze mají stejnou strukturu. V zařízeních, se kterými jsem pracoval, však byla daty naplněna vždy jen jedna z databází, druhá byla prázdná.

Databáze obsahuje několik zajímavých tabulek. Jsou zde uloženy doporučované kontakty (většinou kontakty z Google účtu) v tabulce `suggested_contacts` (výpis 6.4), a v několika tabulkách (`conversation`, `conversation_participants`, `pacticipants` a `messages`) (výpis 6.5) detaily jednotlivých konverzací.

```
CREATE TABLE suggested_contacts (_id INTEGER PRIMARY KEY, gaia_id TEXT,
    chat_id TEXT, name TEXT, first_name TEXT, packed_circle_ids TEXT,
    profile_photo_url TEXT, sequence INT, suggestion_type INT);
```

Výpis kódu 6.4: Tabulka `suggested_contacts`

```
CREATE TABLE conversation_participants (_id INTEGER PRIMARY KEY,
    participant_row_id INT, participant_type INT, conversation_id TEXT,
    sequence INT, active INT, UNIQUE (conversation_id,participant_row_id)
    ON CONFLICT REPLACE, FOREIGN KEY (conversation_id) REFERENCES
    conversations(conversation_id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (participant_row_id) REFERENCES participants(_id));
CREATE TABLE conversations (_id INTEGER PRIMARY KEY, conversation_id TEXT,
    conversation_type INT, latest_message_timestamp INT DEFAULT(0),
    latest_message_expiration_timestamp INT, metadata_present INT,
    notification_level INT, name TEXT, generated_name TEXT, snippet_type
    INT, snippet_text TEXT, snippet_image_url TEXT, snippet_author_gaia_id
    TEXT, snippet_author_chat_id TEXT, snippet_message_row_id INT,
    snippet_status INT, status INT, view INT, inviter_gaia_id TEXT,
    inviter_chat_id TEXT,..., UNIQUE (conversation_id ));
CREATE TABLE messages (_id INTEGER PRIMARY KEY, message_id TEXT,
    message_type INT, conversation_id TEXT, author_chat_id TEXT,
    author_gaia_id TEXT, text TEXT, timestamp INT, status INT, type INT,
    local_url TEXT, remote_url TEXT, attachment_content_type TEXT,
    width_pixels INT, height_pixels INT, stream_id TEXT, image_id TEXT,
```

[16] *Hangouts – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.google.android.talk

```
    album_id TEXT, location_name TEXT, latitude DOUBLE, longitude DOUBLE,
    ..., FOREIGN KEY (conversation_id) REFERENCES conversations(
    conversation_id) ON DELETE CASCADE ON UPDATE CASCADE,UNIQUE (
    conversation_id,message_id) ON CONFLICT REPLACE);
CREATE TABLE participants (_id INTEGER PRIMARY KEY, participant_type INT
    DEFAULT 1, gaia_id TEXT, chat_id TEXT, phone_id TEXT, circle_id TEXT,
    first_name TEXT, full_name TEXT, fallback_name TEXT, profile_photo_url
    TEXT, batch_gebi_tag STRING DEFAULT('-1'), blocked INT DEFAULT(0),
    UNIQUE (circle_id) ON CONFLICT REPLACE, UNIQUE (chat_id) ON CONFLICT
    REPLACE, UNIQUE (gaia_id) ON CONFLICT REPLACE, UNIQUE (phone_id) ON
    CONFLICT REPLACE);
```

Výpis kódu 6.5: Tabulky `conversation`, `conversation_participants`, `pacticipants` a `messages`

Facebook Messenger

Další komunikační program[17]. Celý název balíku je `com.facebook.orca`. Ve složce s databázemi jsou zajímavé především databáze `contacts_db2` a `threads_db2`. První obsahuje tabulku `contacts` (výpis 6.6) ukládající informace o kontaktech ve Facebook Messengeru, ve druhé jsou uložena data proběhlých konverzací (v tabulkách `threads` a `messages` (výpis 6.7)). V tabulce ukládající jednotlivé zprávy jsou informace o odesílateli uloženy přímo jako strukturovaný řetězec obsahující email, id a jméno odesílatele (na rozdíl od předchozí aplikace, kde byly tyto údaje uloženy jako cizí klíče do jiných tabulek).

```
CREATE TABLE contacts (internal_id INTEGER PRIMARY KEY AUTOINCREMENT,
    contact_id TEXT UNIQUE, fbid TEXT, first_name TEXT, last_name TEXT,
    display_name TEXT, small_picture_url TEXT, big_picture_url TEXT,
    huge_picture_url TEXT, small_picture_size INTEGER, big_picture_size
    INTEGER, huge_picture_size INTEGER, communication_rank REAL,
    is_mobile_pushable INTEGER, is_messenger_user TEXT,
    messenger_install_time_ms INTEGER, added_time_ms INTEGER,
    phonebook_section_key TEXT, is_on_viewer_contact_list TEXT, type TEXT,
    link_type TEXT, is_indexed INTEGER, data TEXT );
```

Výpis kódu 6.6: Struktura tabulky `contacts`

```
CREATE TABLE messages (msg_id TEXT PRIMARY KEY, thread_id TEXT, action_id
    INTEGER, text TEXT, sender TEXT, timestamp_ms INTEGER,
    timestamp_sent_ms INTEGER, attachments TEXT, shares TEXT, msg_type
    INTEGER, affected_users TEXT, coordinates TEXT, offline_threading_id
    TEXT, source TEXT, channel_source TEXT, is_non_authoritative INTEGER,
    pending_send_media_attachment STRING, pending_shares STRING,
    pending_attachment_fbid STRING, client_tags TEXT, send_error STRING,
    send_error_message STRING, send_error_timestamp_ms INTEGER, publicity
    TEXT, tracking TEXT );
```

Výpis kódu 6.7: Struktura tabulky `messages`

[17] *Facebook Messenger – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.facebook.orca

Facebook

Aplikace sloužící pro zobrazení stránky Facebook v mobilním zařízení[18]. Celý název balíku aplikace je `com.facebook.katana`. Program používá k uchovávání dat hodně databází, zajímavá data ale obsahuje jen menší část z nich. V databázi `bookmarks` se nachází stejnojmenná tabulka (výpis 6.8), ze které můžeme vyčíst informace o skupinách, stránkách a seznamech přátel. Další dvě zajímavé databáze jsou shodné s předchozím programem.

```
CREATE TABLE bookmarks (_id INTEGER PRIMARY KEY, bookmark_fbid INTEGER,
    bookmark_name TEXT, bookmark_url TEXT, bookmark_icon TEXT, bookmark_pic
     TEXT, bookmark_type TEXT, bookmark_unread_count INTEGER,
    bookmark_client_token TEXT, group_id TEXT, group_name TEXT, group_index
     INTEGER, visible_in_group INTEGER);
```

Výpis kódu 6.8: Struktura tabulky `bookmarks`

Foursquare

Aplikace pro síť Foursquare, která se zaměřuje na sdílení polohy v rámci okruhu přátel[19]. Celý název balíku je `com.joelapenna.foursquared`. Aplikace využívá databázi pouze k uložení seznamu kontaktů. Ten je uložen v souboru `friends.db`. Tato databáze obsahuje stejnojmennou tabulku (výpis 6.9), ze které je možné získat seznam jmen.

```
CREATE TABLE friends(uid TEXT NOT NULL,firstname TEXT,lastname TEXT,
    photoUrl TEXT,photoPrefix TEXT,photoSuffix TEXT,twitterId TEXT,isFriend
     INTEGER,timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,lastActivity
    TEXT,lastActivityId TEXT,lastActivityTimestamp TIMESTAMP DEFAULT
    CURRENT_TIMESTAMP,UNIQUE (uid), PRIMARY KEY (uid));
```

Výpis kódu 6.9: Struktura tabulky `friends`

Keep

Aplikace pro ukládání poznámek[20]. Její celý název je `com.google.android.com`. Databáze aplikace je uložena v souboru `keep.db`. V databázi se nachází velké množství tabulek, užitečná data uchovávají především dvě z nich – `tree_entity` a `list_item` (výpis 6.10). První uchovává názvy poznámek, v druhé jsou uloženy položky jednotlivých poznámek.

```
CREATE TABLE list_item (_id INTEGER PRIMARY KEY AUTOINCREMENT,account_id
    INTEGER NOT NULL,uuid TEXT NOT NULL,server_id TEXT,text TEXT,
    list_parent_id INTEGER NOT NULL,order_in_parent INTEGER NOT NULL
    DEFAULT 0,is_checked INTEGER NOT NULL DEFAULT 0,time_created INTEGER,
    time_last_updated INTEGER,is_dirty INTEGER NOT NULL DEFAULT 0,
    is_deleted INTEGER NOT NULL DEFAULT 0,version INTEGER NOT NULL DEFAULT
    0, base_version TEXT, merge_token TEXT,UNIQUE (account_id, uuid));
```

[18] *Facebook – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.facebook.katana

[19] *Foursquare – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.joelapenna.foursquared

[20] *Google Keep – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.google.android.keep

```
CREATE TABLE tree_entity (_id INTEGER PRIMARY KEY AUTOINCREMENT,account_id
    INTEGER NOT NULL,uuid TEXT NOT NULL,server_id TEXT,type INTEGER NOT
    NULL DEFAULT 0,title TEXT,color_name TEXT,parent_id INTEGER NOT NULL
    DEFAULT 0,order_in_parent INTEGER NOT NULL DEFAULT 0,is_archived
    INTEGER NOT NULL DEFAULT 0,time_created INTEGER,time_last_updated
    INTEGER,is_dirty INTEGER NOT NULL DEFAULT 0,is_deleted INTEGER NOT NULL
     DEFAULT 0,version INTEGER NOT NULL DEFAULT 0, is_trashed INTEGER NOT
    NULL DEFAULT 0, is_graveyard_off INTEGER NOT NULL DEFAULT 0,
    is_graveyard_closed INTEGER NOT NULL DEFAULT 0,
    is_new_list_item_from_top INTEGER NOT NULL DEFAULT 0, base_version TEXT
    ,UNIQUE (account_id, uuid));
```

Výpis kódu 6.10: Struktura tabulek `tree_entity` a `list_item`

Obchod Play a jeho služby

Obchod Google Play slouží k získávání aplikací a je úzce spojen také s aplikací Služby Google Play[21]. Názvy balíků jsou `com.android.vending` pro obchod Play a `com.google.android.gms` pro Služby Google Play.

Obchod Play ukládá data ve třech databázích, jedna ze dvou zajímavých se jmenuje `suggestions.db`. Druhá databáze s užitečnými daty se jmenuje `library.db`. V obou databázích se nachází jedna tabulka. V první jmenované je to tabulka `suggestions` (výpis 6.11), která ukládá historii hledaných řetězců. V druhé je to tabulka `ownership` (výpis 6.12), ve které nalezneme detaily o stažených/zakoupených aplikacích.

```
CREATE TABLE suggestions (_id INTEGER PRIMARY KEY,display1 TEXT UNIQUE ON
    CONFLICT REPLACE,query TEXT,date LONG);
```

Výpis kódu 6.11: Struktura tabulky `suggestions`

```
CREATE TABLE ownership (account STRING, library_id STRING, backend INTEGER,
     doc_id STRING, doc_type INTEGER, offer_type INTEGER, document_hash
    INTEGER, subs_valid_until_time INTEGER, app_certificate_hash STRING,
    app_refund_pre_delivery_endtime_ms INTEGER,
    app_refund_post_delivery_window_ms INTEGER, subs_auto_renewing INTEGER,
     subs_initiation_time INTEGER, subs_trial_until_time INTEGER,
    inapp_purchase_data STRING, inapp_signature STRING, PRIMARY KEY (
    account, library_id, backend, doc_id, doc_type, offer_type));
```

Výpis kódu 6.12: Struktura tabulky `ownership`

Služby Google Play ukládají data ve více databázích. Nejzajímavější je databáze `plus-contacts.db`, ze které můžeme získat seznam kontaktů na **Google+**. Informace o jednotlivých kontaktech dostaneme spojením údajů z tabulek `people`, `phones` a `postal_address` 6.13.

```
CREATE TABLE people (_id INTEGER PRIMARY KEY AUTOINCREMENT,owner_id INTEGER
     NOT NULL,qualified_id TEXT NOT NULL,gaia_id TEXT,v2_id TEXT NOT NULL,
    name TEXT,given_name TEXT,family_name TEXT,middle_name TEXT,
```

[21] Stránka aplikace na Google Play – https://play.google.com/store/apps/details?id=com.google.android.gms

```
    profile_type INTEGER NOT NULL,sort_key TEXT,sort_key_last_name TEXT,
    sort_key_irank TEXT,avatar TEXT,tagline TEXT,blocked INTEGER NOT NULL
    DEFAULT 0,etag TEXT,last_modified INTEGER NOT NULL DEFAULT 0,
    invisible_3p INTEGER NOT NULL DEFAULT 0,in_viewer_domain INTEGER NOT
    NULL DEFAULT 0 ,UNIQUE (owner_id,qualified_id),FOREIGN KEY (owner_id)
    REFERENCES owners(_id) ON DELETE CASCADE);
CREATE TABLE phones (_id INTEGER PRIMARY KEY AUTOINCREMENT,owner_id INTEGER
     NOT NULL,qualified_id TEXT NOT NULL,phone TEXT NOT NULL,type INTEGER
    NOT NULL,custom_label TEXT,FOREIGN KEY (owner_id,qualified_id)
    REFERENCES people(owner_id,qualified_id) ON DELETE CASCADE);
CREATE TABLE postal_address (_id INTEGER PRIMARY KEY AUTOINCREMENT,owner_id
     INTEGER NOT NULL,qualified_id TEXT NOT NULL,postal_address TEXT NOT
    NULL,type INTEGER NOT NULL,custom_label TEXT,FOREIGN KEY (owner_id,
    qualified_id) REFERENCES people(owner_id,qualified_id) ON DELETE
    CASCADE);
```

Výpis kódu 6.13: Struktura tabulek `people`, `phones` a `postal_address`

Vyhledávání

Aplikace pro vyhledávání, celý název je `com.android.quicksearchbox`. Data ukládá v jediné databázi – `qsb-log.db`. V databázi jsou zajímavé dvě tabulky – `shortcuts` (výpis 6.14) a `clicklog` (výpis 6.15), ze kterých můžeme získat historii vyhledávání.

```
CREATE TABLE shortcuts (intent_key TEXT NOT NULL COLLATE UNICODE PRIMARY
    KEY, source TEXT NOT NULL, source_version_code INTEGER NOT NULL, format
     TEXT, title TEXT, description TEXT, description_url TEXT, icon1 TEXT,
    icon2 TEXT, intent_action TEXT, intent_component TEXT, intent_data TEXT
    , intent_query TEXT, intent_extradata TEXT, shortcut_id TEXT,
    spinner_while_refreshing TEXT, log_type TEXT, custom_columns TEXT);
```

Výpis kódu 6.14: Struktura tabulky `shortcuts`

```
CREATE TABLE clicklog ( _id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    intent_key TEXT NOT NULL COLLATE UNICODE REFERENCES shortcuts(
    intent_key), query TEXT, hit_time INTEGER,corpus TEXT);
```

Výpis kódu 6.15: Struktura tabulky `clicklog`

YouTube

Oficiální aplikace pro video portál YouTube[22]. Název balíku je `com.google.android.youtube`. Aplikace využívá tři databáze. Užitečná data se však nachází pouze v databázi `history.db`. V této databázi je jediná tabulka – `suggestions` (výpis 6.16), ve které jsou uloženy v aplikaci vyhledávané řetězce.

```
CREATE TABLE suggestions (_id INTEGER PRIMARY KEY,display1 TEXT UNIQUE ON
    CONFLICT REPLACE,display2 TEXT,query TEXT,date LONG);
```

Výpis kódu 6.16: Struktura tabulky `suggestions`

[22] *YouTube – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.google.android.youtube

Mapy.cz

Aplikace s mapovými podklady od Seznamu[23]. Celý název balíku je `cz.seznam.mapy`. Ze zajímavých dat ukládá aplikace do databáze pouze historii vyhledávání, ta se nachází v tabulce `SearchHistoryItem` (výpis 6.17) v databázi `maps_cz_database.db`.

```
CREATE TABLE SearchHistoryItem(_id INTEGER PRIMARY KEY AUTOINCREMENT, title
    TEXT UNIQUE ON CONFLICT REPLACE NOT NULL, subtitle TEXT, poiId INTEGER
    NOT NULL, icon TEXT, phoneNumber TEXT, positionX INTEGER NOT NULL,
   positionY INTEGER NOT NULL, zoom INTEGER NOT NULL, timeStamp INTEGER
   NOT NULL);
```

Výpis kódu 6.17: Struktura tabulky `SearchHistoryItem`

Google Mapy

Aplikace s mapovými podklady od Google[24]. Celý název balíku aplikace je `com.google.android.apps.maps`. Stejně jako aplikace Mapy.cz ukládá aplikace Google Mapy historii vyhledávání. Ta se nachází v databázi s názvem `search_history.db` v tabulce `suggestions` (výpis 6.18).

```
CREATE TABLE suggestions (_id INTEGER PRIMARY KEY AUTOINCREMENT, data1 TEXT
   , singleResult INTEGER, displayQuery TEXT, latitude INTEGER DEFAULT
   200000000, longitude INTEGER DEFAULT 200000000, timestamp INTEGER);
```

Výpis kódu 6.18: Tabulka `suggestions`

Schoology

Aplikace pro portál Schoology[25]. Celý název balíku je `com.schoology.app`. Aplikace ukládá data pouze v jedné databázi – `schoology.db`. V této databázi jsou 4 tabulky, nejzajímavější je tabulka `users` (výpis 6.19), ve které se nachází seznam uživatelů, kteří mají zapsaný stejný kurz.

```
CREATE TABLE users (user_id INTEGER PRIMARY KEY,user_name TEXT,user_img_url
    TEXT,post_self_update INTEGER,post_message INTEGER,user_img BLOB);
```

Výpis kódu 6.19: Struktura databáze `schoology.db`

Další aplikace

Pro operační systém Android existuje velmi mnoho aplikací (jenom na Google Play jich je přes jeden milion[26]). Výše byly uvedeny pouze takové, které aktivně používám, a proto bylo pravděpodobné, že budou obsahovat nějaká soukromá data. Z dalších aplikací bych

[23] *Mapy.cz – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=cz.seznam.mapy

[24] *Mapy – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.google.android.apps.maps

[25] *Schoology – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.schoology.app

[26] http://www.appbrain.com/stats/number-of-android-apps

zmínil ještě aplikace sociální sítě Twitter[27], jejíž adresář s daty obsahuje několik souborů s databázemi různých jmen, ale podobné struktury. V mém případě byla data uložena v databázích s názvem `119355918-2.db` a `119355918-10.db`. V databázích se nacházelo poměrně velké množství dat, žádná z nich mi ale nepřišla příliš citlivá, jednalo se převážně o lokální uložení některých zobrazených tweetů, které jsou veřejné.

6.2 Součásti aplikace

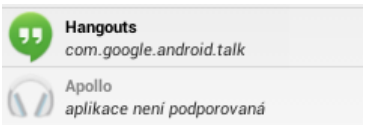
Aplikace DPForensic je tvořena jednou aktivitou, která umožňuje interakci s uživatelem. Většina výpočtů je implementována v asynchroních úlohách, mimo hlavní vlákno uživatelského rozhraní, bylo tak docíleno dobré odezvy programu. V tabulce 6.1 je znázorněna architektura aplikace.

Balík	Popis
main	Spuštění a zobrazení aplikace. V tomto balíku je implementace aktivity a pomocných adaptérů a fragmentů.
func	Hlavní logika aplikace. V tomto balíku jsou implementace všech funkcí aplikace.
data	Reprezentace dat aplikací. Instance tříd z tohoto balíku reprezentují získaná data z jednotlivých aplikací.

Tabulka 6.1: Přehled architektury aplikace

Hlavní obrazovka

Hlavní obrazovka programu je implementována v třídě `ListAppsActivity.java` (dostupná ve složce `main`). Aktivita se spustí po startu aplikace a ihned při vytváření (v metodě `onCreate`) spustí načítání seznamu aplikací. Načítání probíhá v novém vlákně a uživatel je o jeho dokončení informován zmizením dialogu o průběhu činnosti. Po načtení všech aplikací uživatel vidí přehledný seznam, ve kterém jsou vizuálně oddělené podporované a nepodporované aplikace (obrázek 6.1). Součástí hlavní obrazovky je menu, které umožňuje zobrazit nápovědu aplikace a také volby pro další činnosti aplikace.



Obrázek 6.1: Rozdíl v zobrazení podporované a nepodporované aplikace

Funkce

Ve složce `func` jsou implementovány jednotlivé funkce, které aplikace poskytuje.

[27] *Twitter – Aplikace pro Android ve službě Google Play* – https://play.google.com/store/apps/details?id=com.twitter.android

- `HandleDir.java` – pomocná třída, která obsahuje pouze metody pro práci se složkami – vytváření a mazání.
- `MakeXML.java` – třída obsahuje metody pro uložení získaných dat v čitelném formátu XML. Pro různé typy dat existují specifické metody. Vytvořené XML soubory jsou ukládány do složky na paměťové kartě, pojmenovány jsou názvem aplikace, ze které pochází data.
- `ParseData.java` – hlavní třída pro získávání dat, obsahuje metody pro jednotlivé podporované aplikace (ze sekce 6.1). Metody pracují s vestavěným API (SMS, kontakty, kalendář) nebo s kopiemi datových složek aplikací, které se ukládají do dočasné složky na paměťové kartě. Získaná data jsou poté pomocí výše zmíněných metod uložena do XML souborů.
- `RootAccess.java` – tato třída obsahuje metody, které umožňují spouštět příkazy pod uživatelem `root`. Pokud je potřeba vykonat více příkazů s nutností vyšších práv, jsou vykonány v jednom procesu, aby se zamezilo nutnosti častého povolování oprávnění při práci s aplikací. Metody v této třídě vznikly s pomocí příspěvku na fóru xda-developers[28].
- `SuppAppTest.java` – další pomocná třída. Obsahuje metody, které slouží k testování zda-li se jedná o podporovanou aplikaci či nikoliv. Aplikace jsou rozlišovány pomocí názvu svých balíků.

Data

Ve složce `data` jsou třídy, reprezentující data jednotlivých aplikací, pomocí instancí těchto tříd jsou získaná data předána metodám v třídě `MakeXML`, které je uloží v textové podobě v XML souboru. Příklad struktury XML souboru je na výpisu 6.20.

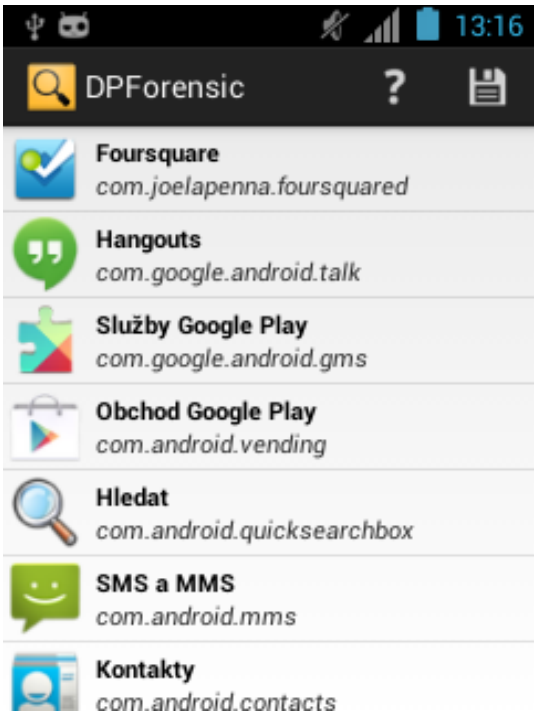
```
<seznam_hesel>
        <stranka>
                <www>https://cas.fit.vutbr.cz</www>
                <username>xhanya02</username>
                <password>tajneHeslo</password>
        </stranka>
        <stranka>
                <www>https://m.facebook.com</www>
                <username>m.hanyas@gmail.com</username>
                <password>tajneHeslo</password>
        </stranka>
        <stranka>
                <www>https://wifigw.cis.vutbr.cz</www>
                <username>xhanya02</username>
                <password>tajneHeslo</password>
        </stranka>
</seznam_hesel>
```

Výpis kódu 6.20: Část XML souboru s daty prohlížeče

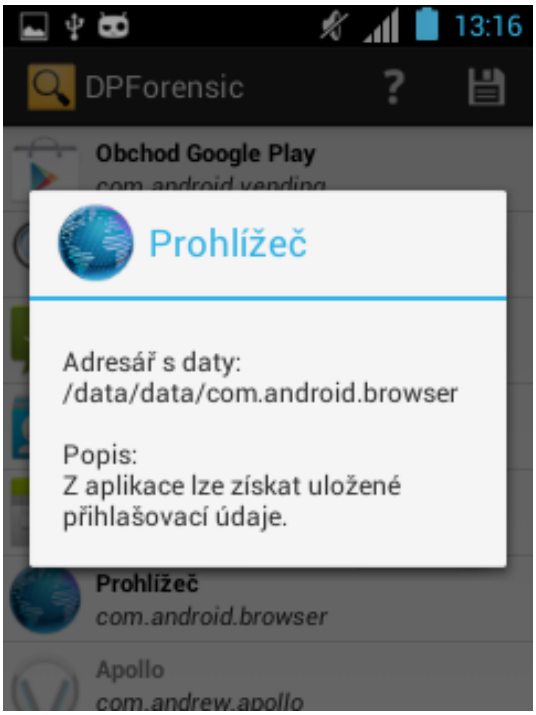
[28] *xda-developers – Writing an android root application* – http://forum.xda-developers.com/showpost.php?p=4528387&postcount=14

6.3 Uživatelské rozhraní

Uživatelské rozhraní bylo přizpůsobeno jednoduchému, intuitivnímu a rychlému ovládání. Oproti návrhu (kapitola 5.2) má aplikace pouze jednu obrazovku (seznam nainstalovaných aplikací, obrázek 6.2) a většina voleb se odehrává pomocí menu. V hlavní menu (obrázek 6.5) jsou volby pro práci se všemi podporovanými aplikacemi, v kontextovém menu (obrázek 6.4) jednotlivých aplikací je možné vybrat činnost pro jednu konkrétní vybranou aplikaci. Kontextové menu se zobrazuje při dlouhém stisku řádku se jménem aplikace, při krátkém stisku je zobrazen dialog s detailnějším popisem aplikace (obrázek 6.3).



Obrázek 6.2: Seznam aplikací



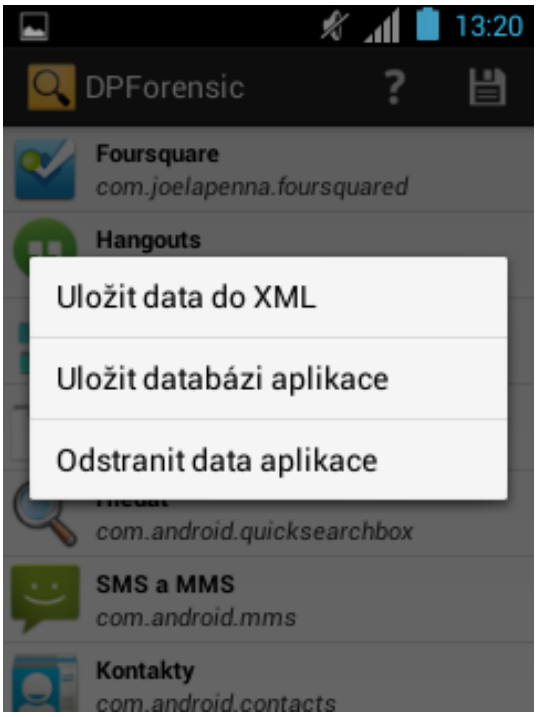
Obrázek 6.3: Detail vybrané aplikace

6.4 Optimalizace pro mobilní zařízení

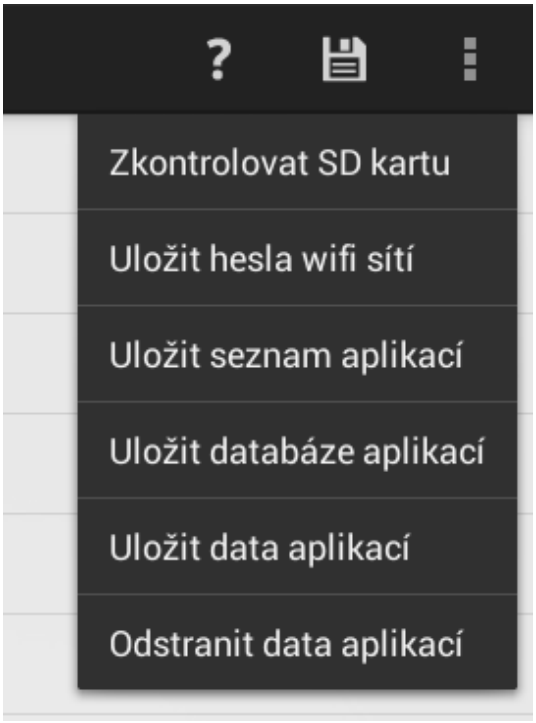
Tento odstavec vychází z [2], kde je uveden souhrn doporučení pro programátorské praktiky při vývoji aplikace běžící na mobilním zařízení s OS Android. Uvedu zde pouze techniky, které jsem přímo využil.

Nevytváření nadbytečných objektů

Zařízení s OS andtroid využívají ke správě paměti *garbage collector*. Ten slouží k uvolňování paměti odstraňováním nepotřebných objektů. Jeho spuštění ale znamená snížení výkonu zařízení, je tedy velice vhodné, aby byl spouštěný co možná nejméně. Jelikož existují i zařízení s velmi malou operační pamětí a současně malým výkonem, spuštění garbage collectoru má na výkon obrovský vliv. Jedním z doporučení je tedy vyvářet co možná nejméně nových objektů a pokud je to možné, znovu používat objekty již vytvořené. Tím zabráníme zbytečnému navyšování nároků na paměť a tedy i nutnosti paměť uvolňovat garbage collectorem.



Obrázek 6.4: Kontextové menu



Obrázek 6.5: Hlavní menu aplikace

Používání statických metod

Volání statické metody je o 15-20% rychlejší.

Cyklus for-each

U kolekcí, které implementují rozhraní `Iterable`[29], a u polí je možné používát rozšířený cyklus for přes všechny prvky pole (kolekce) – takzvaný for-each cyklus, který je velice přehledný a zároveň nejrychlejší metodou procházení seznamu. Proto je tato konstrukce využita ve většině případů, kde to bylo možné.

BufferedWriter

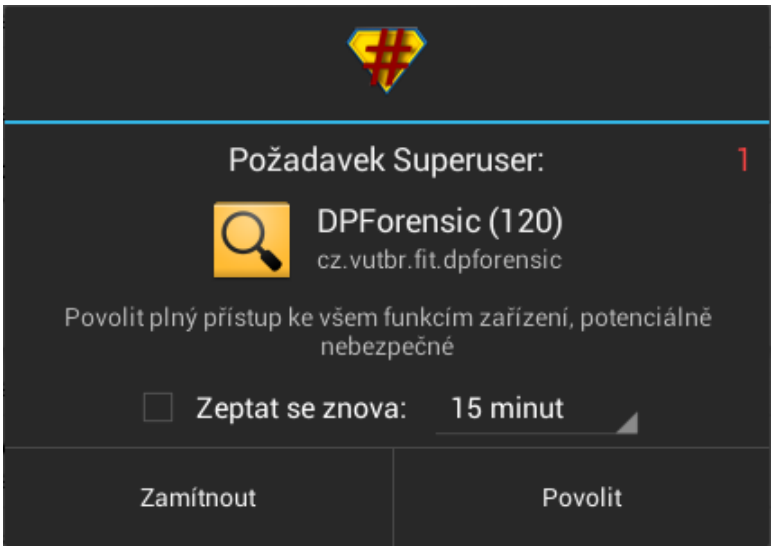
Pro zápis do souboru je využit `BufferedWriter`[30], který minimalizuje režii systému spojenou s přímým zápisem do souboru. Jelikož při vytváření XML postupně připisuji relativně krátké sekvence znaků, je toto řešení výhodné i na úkor větším nárokům na paměť.

6.5 Požadavky

Aplikace ke svému spuštění a bezproblémovému běhu vyžaduje OS Android minimálně ve verzi 4.0 (API verze 14[31]). Aplikaci lze spustit i na nižší verzi, nebude však možné využít získání dat z kalendáře, pro které je využíváno funkcí právě z API verze 14.

[29] *Android Developers* – http://developer.android.com/reference/java/lang/Iterable.html
[30] *Android Developers* – http://developer.android.com/reference/java/io/BufferedWriter.html
[31] *Android 4.0 APIs* – http://developer.android.com/about/versions/android-4.0.html

Aplikaci je vhodné spouštět na zařízení, na kterém byl proveden *root*, a aplikaci tato práva povolit (například obrázek 6.6 ukazuje povolování pomocí aplikace SuperSU[32]). Při spuštění na zařízení s běžnými právy nebo při nepovolení práv *root* dojde ke snížení funkčnosti – aplikace nebude moci přistupovat k datům nainstalovaných aplikací.



Obrázek 6.6: Povolení práv aplikací SuperSU

6.6 Testování

Pro testování během implementace jsem používal dvě zařízení – mobilní telefon a tablet. Jednalo se o mobilní telefon Samsung Galaxy Mini[33] s ROM CyanogenMod ve verzi 10.1.6 (upravený Android ve verzi 4.2.2) a tablet Ainol Novo 7 Crystal s originálním OS Android ve verzi 4.1.1. Obě zařízení měla proveden *root*.

Pro ověření funkčnosti i na dalších zařízeních jsem aplikaci nainstaloval na tablet Google Nexus 7 s operačním systémem Android verze 4.4.2 (v době psaní tohoto textu aktuální verze). Na tomto zařízení nebyl proveden *root*, proto aplikace fungovala pouze v omezené variantě, kdy nebylo možné získat data z nainstalovaných aplikací. Aplikace byla také vyzkoušena na telefonu HTC Desire S[34] s operačním systémem Android ve verzi 4.0.

Na vyzkoušených zařízeních bylo možné aplikaci bez problémů spustit a pracovat s ní. Pokud byl aplikaci umožněn *root* přístup, byla bez problémů přečtena data podporovaných aplikací. Při pokusu o vymazání dat z nainstalovaných aplikací nastal problém na tabletu Ainol Novo, na kterém nebylo možné modifikovat soubory ve vnitřní paměti (ani při pokusech s jinými aplikacemi). Jednalo se tak pravděpodobně o špatně provedený *root*.

[32] *SuperSU* – https://play.google.com/store/apps/details?id=eu.chainfire.supersu

[33] *Technické specifikace telefonu Samsung Galaxy Mini* – http://www.gsmarena.com/samsung_galaxy_mini_s5570-3725.php

[34] *Technické specifikace telefonu HTC Desire S* – http://www.gsmarena.com/htc_desire_s-3776.php

Kapitola 7

Podklady pro výuku

V této kapitole popíši provedení forenzní analýzy konkrétního mobilního telefonu[1] několika různými nástroji. Součásti jsou demonstrační úlohy (materiály do laboratoří), které se nachází v příloze B a demonstrační videa, která jsou uložena na přiloženém nosiči.

Součástí je také forenzní posudek, exportovaný z programu Oxygen Forensic® Suite. Tento posudek je zpracován pro obraz zařízení ze stránek Oxygen Forensic®[2]. Tento posudek je také uložen na přiloženém nosiči.

7.1 Postupy provedení forenzní analýzy

7.1.1 viaExtract

Jelikož je viaExtract distribuován jako obraz virtuálního disku, pro spuštění budeme potřebovat vytvořit virtuální počítač, například pomocí programu *VirtualBox*[3] nebo *VMware Player*[4]. Do vytvořeného počítače přidáme stažený disk s nainstalovaným Ubuntu a programem viaExtract. Následně stačí připojit zařízení, které chceme analyzovat, stisknout ikonu *New Case* a pokračovat podle instrukcí forenzního programu.

Do analyzovaného zařízení je během získávání dat nainstalována drobná aplikace (AFLogical OSE, obrázek 7.1), kterou není možné odmítnout. Na rozdíl od dále popisovaného MOBILedit! Forensic je tato aplikace po získání dat ze zařízení automaticky odstraněna.

Po dokončení analýzy, je uživateli zobrazen přehled získaných dat. Jednotlivé položky přehledu je možné tisknout nebo exportovat ve formátu `CSV`. Kompletní zprávu o získaných datech je možné exportovat do `PDF`.

7.1.2 MOBILedit! Forensic

Po spuštění nástroje MOBILedit! Forensic můžeme procházet dříve vytvořené analýzy připojených zařízení, nebo přidat zařízení nové. Po stisknutí příslušné ikony je spuštěn průvodce připojení zařízení, který umožňuje výběr druhu připojení a výběr dat, které z připojeného zařízení chceme získat. Průvodce také informuje o nutnosti zapnutí režimu *Ladění USB* a požádá o instalaci aplikace do zařízení (ME! Forensic Connector, obrázek 7.2). Ta

[1] Samsung Galaxy Mini, Android 4.2.2 (CyanogenMod 10.1.6)

[2] Samsung Galaxy Mini ze stránek *Oxygen Forensic® Suite – Demo Backups* – `http://www.oxygen-forensic.com/en/download/devicebackups`

[3] *Oracle VM VirtualBox* – `https://www.virtualbox.org/`

[4] *VMware Player Plus: Easiest Way to Run a Virtual Machine* – `http://www.vmware.com/cz/products/player/`

Obrázek 7.1: viaExtract – aplikace nainstalovaná do zařízení

není nezbytná, ale bez ní jsou možnosti analýzy velice omezené (je možné pouze procházet souborový systém). Po nastavení všech parametrů je spuštěna samotná analýza, která trvá různě dlouho v závislosti na zvolených detailech a velikosti paměti zařízení. Po ukončení získávání dat je možné tato data procházet jednotlivě, popřípadě (v placené verzi) vyexportovat souhrnnou zprávu.

MOBILedit! Forensic také umožňuje vytvořit obraz paměti připojeného zařízení, stačí stisknout příslušnou ikonu a potvrdit, že je vložena funkční paměťová karta s dostatkem místa, na kterou bude vytvořený obraz uložen.

7.1.3 Oxygen Forensic® Suite

Prvním krokem při připojení zařízení je spuštění nástroje Oxygen Forensic® Extractor, který připojené zařízení detekuje. Pokud automatická detekce selže, je možné zařízení vybrat ručně z nabízeného seznamu. Seznam podporovaných zařízení je rozsáhlý[5], ale lze v něm snadno vyhledávat. Po připojení zkoumaného zařízení jsou nabídnuty dvě možnosti – vybrat metodu extrakce dat automaticky podle konkrétního připojeného modelu, nebo zvolit metodu ručně. Na mnou testovaném mobilním telefonu automaticky zvolená

[5] *Oxygen Forensic® Suite – Supported devices* – http://www.oxygen-forensic.com/en/compare/devices

Obrázek 7.2: MOBILedit! Forensic – aplikace nainstalovaná do zařízení

metoda nezískala mnoho dat. Při ručním výběru metody získání dat, jsou uživateli nabídnuty následující tři postupy:

- Vytvoření obrazu paměti telefonu a jeho následná analýza – vyžaduje root přístup,
- získání dat z telefonu na základě funkce zálohování přes ADB – dostupné pro Android 4.0 a vyšší,
- logická analýza – do telefonu je nainstalovaná aplikace OxyAgent (obrázek 7.3), která umožňuje z telefonu získat konkrétní data (adresář, SMS zprávy, kalendář...).

První dvě metody v mém případě nebyly příliš úspěšné, třetí metoda proběhla v pořádku. Po získání dat z telefonu je možné okamžitě vytisknout přehlednou zprávu o získaných datech, popřípadě získaná data otevřít v programu Oxygen Forensic® Suite. Ten ovšem ve verzi zdarma (Standard) mimo prohlížení získaných dat, neumožňuje mnoho dalších operací. Pokročilejší placená verze nabízí různé agregace a zobrazování dat v časových intervalech nebo jiných souvislostech.

Obrázek 7.3: Oxygen Forensic® Suite – aplikace nainstalovaná do zařízení

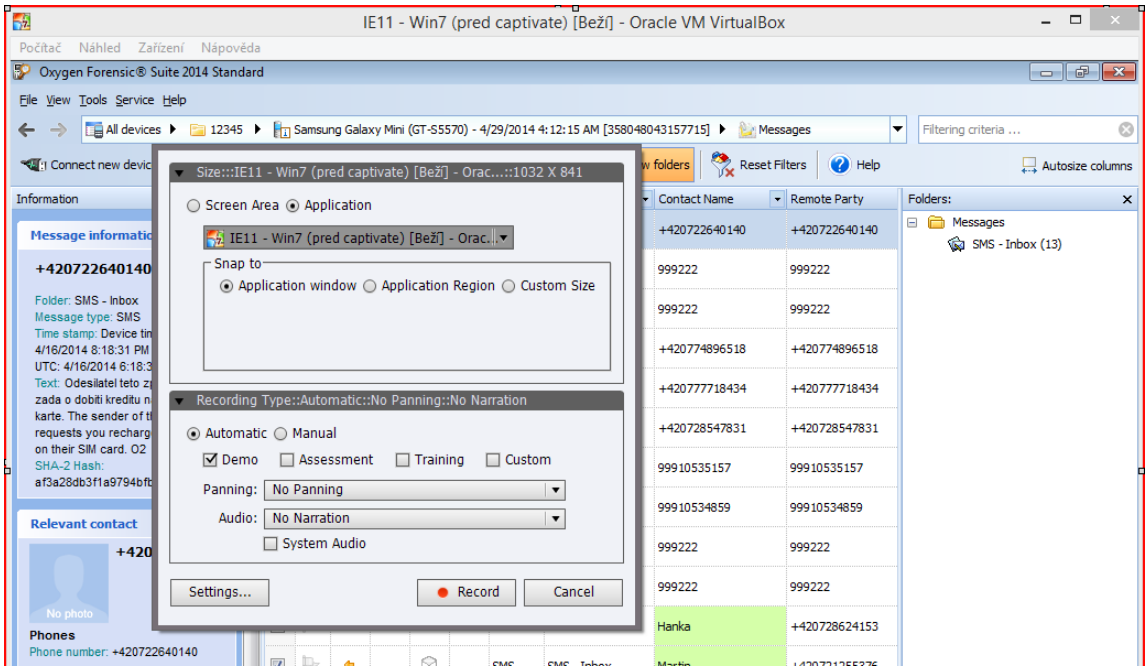
7.2 Demonstrační videa

Pro zachycení a publikování demonstrovaných úloh byl využit program *Adobe Captivate 7*[6]. Tento program umožňuje zachycení činnosti konkrétního okna nebo celé plochy. Zaznamenávat se může kompletní dění, nebo jen významné akce. Zvolil jsem druhou variantu, kdy jsem zaznamenával obsah okna virtuálního počítače (obrázek 7.4) a jednotlivé akce poté v programu doplnil komentářem, popisujícím dění na obrazovce a jeho smysl. Uživatelské rozhraní editoru je ukázáno na obrázku 7.5. Program umožňuje výsledné video exportovat do několika formátů a přímo z programu lze výsledné video publikovat na server YouTube[7] nebo do e-learningového systému. Já jsem zvolil variantu exportování videa ve formátu MP4[8]. Videa se nachází na přiloženém nosiči.

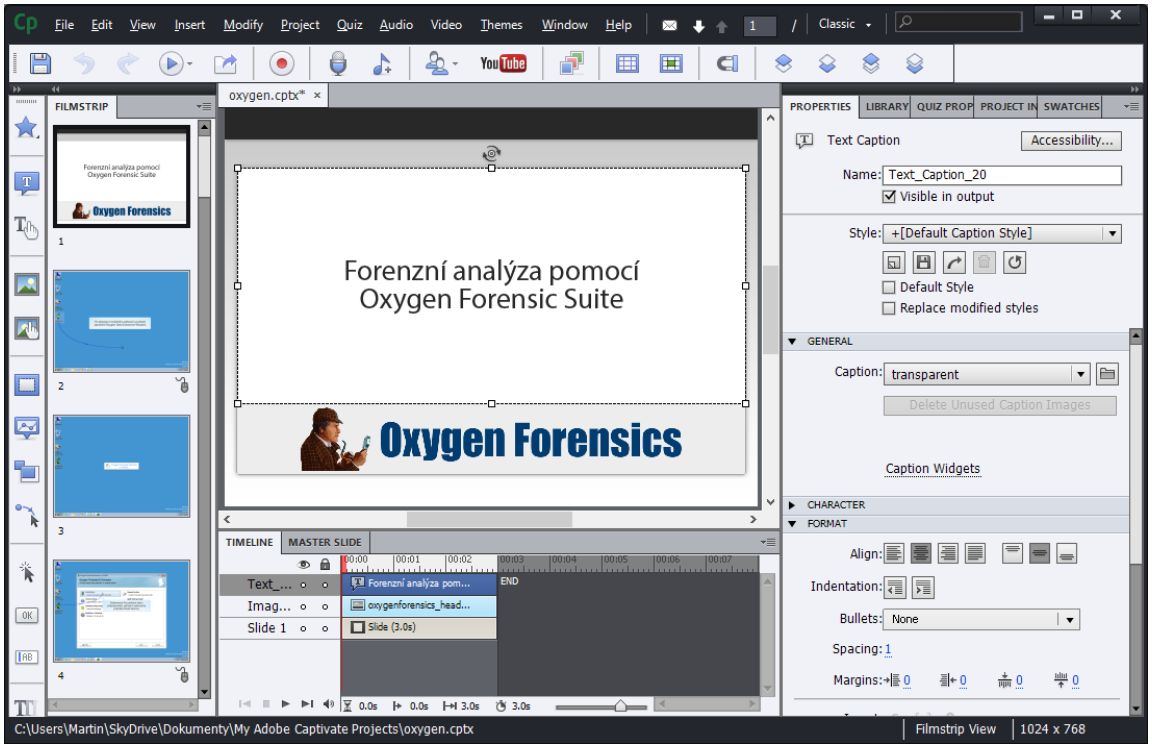
[6] *Screen capture software, E-learning software, HTML5 Publishing, mLearning — Adobe Captivate 7* – http://www.adobe.com/cz/products/captivate.edu.html

[7] *YouTube* – http://www.youtube.com/

[8] ISO/IEC 14496-14 *Information technology – Coding of audio-visual objects – Part 14: MP4 file format*



Obrázek 7.4: Adobe Captivate 7– nahrávání videa



Obrázek 7.5: Adobe Captivate 7– editování videa

Kapitola 8

Závěr

V této diplomové práci jsem se zaměřil na forenzní analýzu zařízení s operačním systémem Android. Prozkoumal jsem dostupné nástroje a pomocí některých provedl forenzní analýzu vybraného zařízení. V této části vznikly demonstrační úlohy a videa, které je možné použít jako podklady pro výuku.

Dále jsem se zaměřil na návrh a implementaci aplikace, která ze zařízení s operačním systémem Android získá informace o majiteli. Po zkušenostech s aplikacemi pro osobní počítače, které jsoum využil pro tvorbu demonstračních úloh, jsem se rozhodl vytvořit aplikaci přímo pro mobilní zařízení. Aplikace umožňuje získat data z několika aplikací a je možné ji dále rozšiřovat o podporu dalších. V současnosti je implementována podpora pro aplikace zmíněné v kapitole 6.1.

S využitím literatury a poznatků získaných při tvorbě vlastní aplikace jsem zjistil, že operační systém Android a aplikace určené pro tento sytém, při ukládání dat spoléhají většinou pouze na systém práv, který umožňuje v defaultním nastavení aplikaci číst pouze svá data, případně přes poskytnutá API přistupovat k datům z jiných aplikací. Pro operační systém Android jsou však známé postupy, které umožňují získat práva uživatele *root* (tyto postupy jsou popsány v kapitole 3.3), který umožňuje přístup ke všem datům v paměti zařízení. Na jednu stranu, tento fakt usnadňuje práci forenzním specialistům, na druhou stranu při ztrátě nebo odcizení mobilního zařízení má případný nový majitel snadný přístup k osobním datům předchozího majitele. Při práci s mobilním zařízením s OS Android je důležité mít tento fakt na mysli a případně provést některé kroky ke zvýšení zabezpečení (šifrování úložiště, povolené vzdálené vymazání zařízení...).

Literatura

[1] *Android Debug Bridge* [online]. [cit. 5. prosince 2013]. Dostupné na: <http://developer.android.com/tools/help/adb.html>.

[2] *Performance Tips* [online]. [cit. 1. května 2014]. Dostupné na: <http://developer.android.com/training/articles/perf-tips.html>.

[3] *Permissions* [online]. [cit. 5. prosince 2013]. Dostupné na: <http://developer.android.com/guide/topics/security/permissions.html>.

[4] *UFED TOUCH ULTIMATE* [online]. [cit. 8. prosince 2013]. Dostupné na: <http://www.ufed.cz/ufed/3-PRODUKTY/4-UFED-TOUCH-ULTIMATE>.

[5] *Android System Architecture* [online]. 2012 [cit. 21. prosince 2013]. Dostupné na: <http://commons.wikimedia.org/wiki/File:Android-System-Architecture.svg/>.

[6] *Cellebrite - Mobile Forensics* [online]. 2013 [cit. 8. prosince 2013]. Dostupné na: <http://www.cellebrite.com/mobile-forensics>.

[7] Cannon, T. *Android Reversing* [online]. 2010 [cit. 11. května 2014]. Dostupné na: <http://thomascannon.net/projects/android-reversing/>.

[8] Casey, E. *Digital evidence and computer crime: forensic science, computers and the Internet*. 3. vyd. Amsterdam: Elsevier, 2011. 807 s. ISBN 978-0-12-374268-1.

[9] Hoog, A. *Android forensics*. Amsterdam: Elsevier, 2011. 372 s. ISBN 978-1-59749-651-3.

[10] Hoog, A. *Mobile security, forensics & malware analysis with Santoku Linux*. 2013. Prezentováno na AppSec USA 2013. Dostupné na: <http://www.slideshare.net/dleyanlin/via-forensics-appsecusanov2013>.

[11] Kadlec, J. *Forenzní analýza unixových systémů*. Hradec Králové: Univerzita Hradec Králové, 2006. Diplomová práce.

[12] Misra, A. a Dubey, A. *Android Security: Attacks and Defenses*. Amsterdam: Elsevier, 2011. 372 s. ISBN 978-1-59749-651-3.

[13] Mysliveček, D. *Krátké ohlédnutí za historií Androidu* [online]. 2013 [cit. 5. prosince 2013]. Dostupné na: <http://www.svetandroida.cz/kratke-ohlednuti-za-historii-androidu-201305>.

[14] Tyler, J. a Verduzco, W. *XDA Developers' Android hacker's toolkit*. Chichester: John Wiley, 2012. 175 s. ISBN 978-1-119-95138-4.

Dodatek A

Obsah CD

- Forenzní posudek (z programu Oxygen Forensic® Suite),
- demonstrační videa (z linuxové distribuce Santoku a programu Oxygen Forensic® Suite),
- demonstrační úlohy (z programů viaExtract, MOBILedit! Forensic a Oxygen Forensic® Suite),
- diplomová práce v PDF a zdrojové soubory LaTeX,
- zdrojové soubory aplikace DPForensic a instalační balíček.

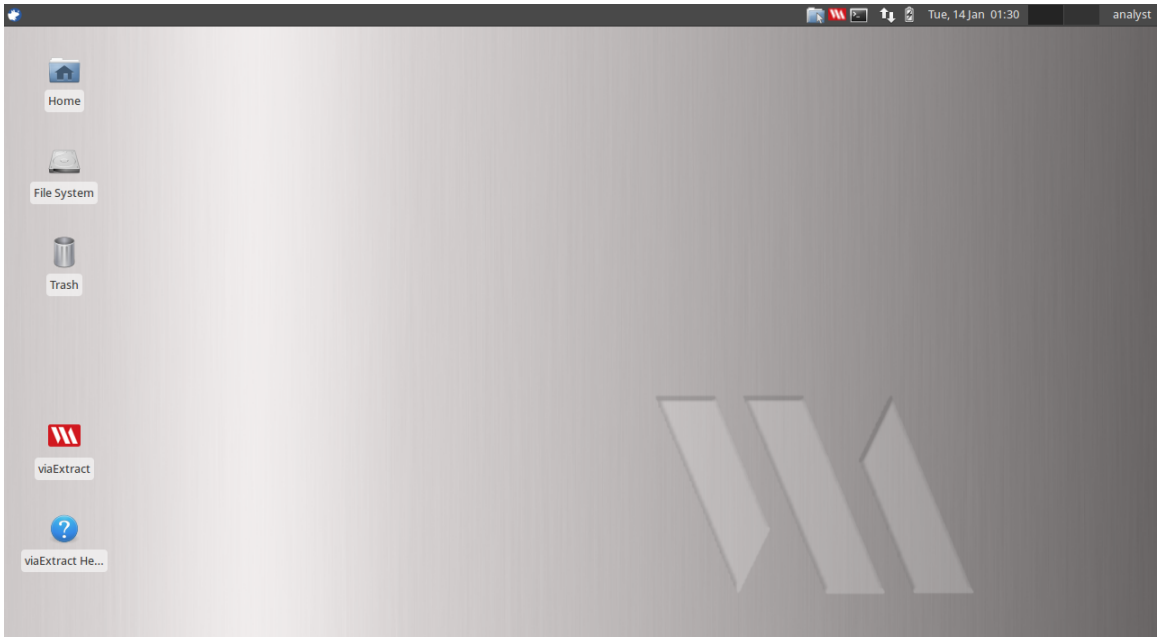
Dodatek B

Demonstrační úlohy

B.1 viaExtract

Instalace a spuštění

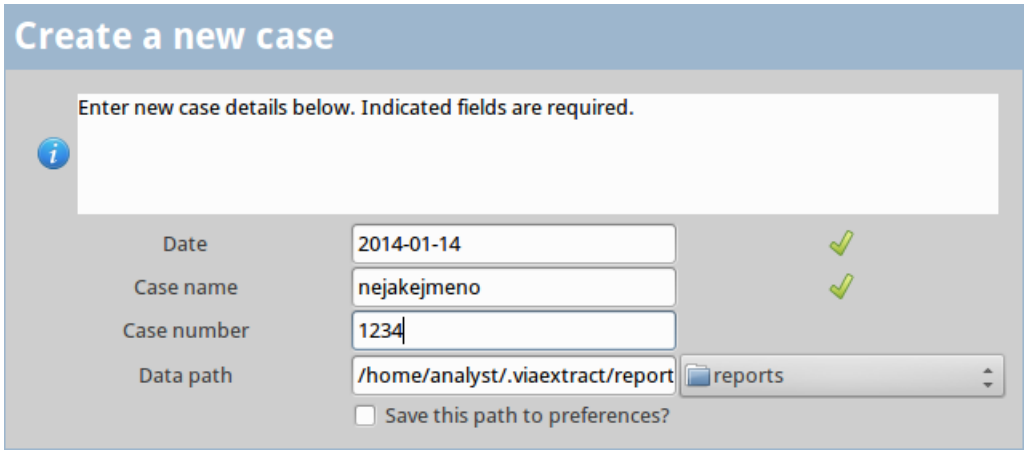
Program viaExtract je distribuován na virtuálním obrazu počítače, kde je tento software nainstalován a nakonfigurován. Instalace tedy spočívá pouze ve spuštění tohoto virtualizovaného počítače. Po přihlášení do systému můžeme okamžitě spustit forenzní program. Při každém spuštění verze bez zavedené licence je nutné přepnout program do DEMO módu.



Obrázek B.1: Stav počítače po přihlášení do systému

Přidání nového případu

Pro vytvoření nového případu klikněte na tlačítko *New*. V průvodci vyplňte potřebné údaje (povinné jsou označeny).



Obrázek B.2: Vytvoření nového případu

Přidání zkoumaného zařízení

Dalším krokem je přidání nového zařízení do vytvořeného případu. Připojte zkoumané zařízení a opět klikněte na tlačítko *New*. Tentokrát se spustí průvodce přidáním zařízení. Opět vyplňte potřebné údaje.

Následuje výběr činnosti, kterou chcete provést. V DEMO módu je aktivní pouze extrakce dat ze zařízení.

Obrázek B.3: Výběr činnosti

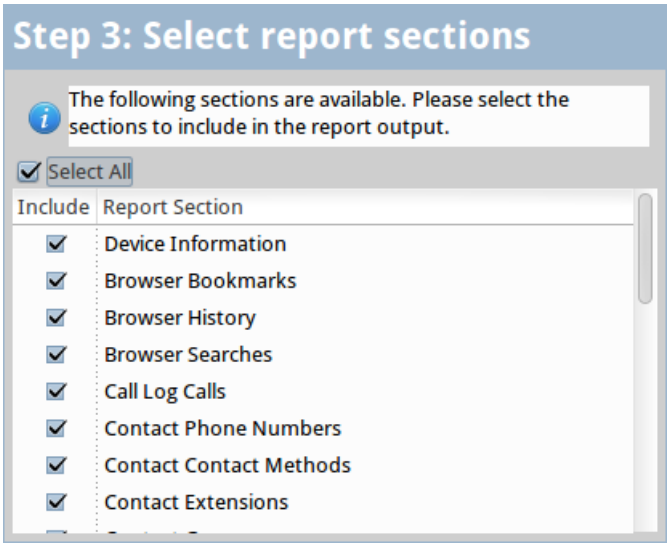
Nastavení zařízení

Po spuštění extrakce jsou zobrazeny úkony, které je nutné provést na připojeném zařízení. Po jejich provedení je možné pokračovat. Do připojeného telefonu je nainstalována malá aplikace, které zajišťuje extrakci dat.

Obrázek B.4: Nastavení analyzovaného zařízení

Extrakce dat

Dále je možné zvolit, která data budou ze zařízení získána.



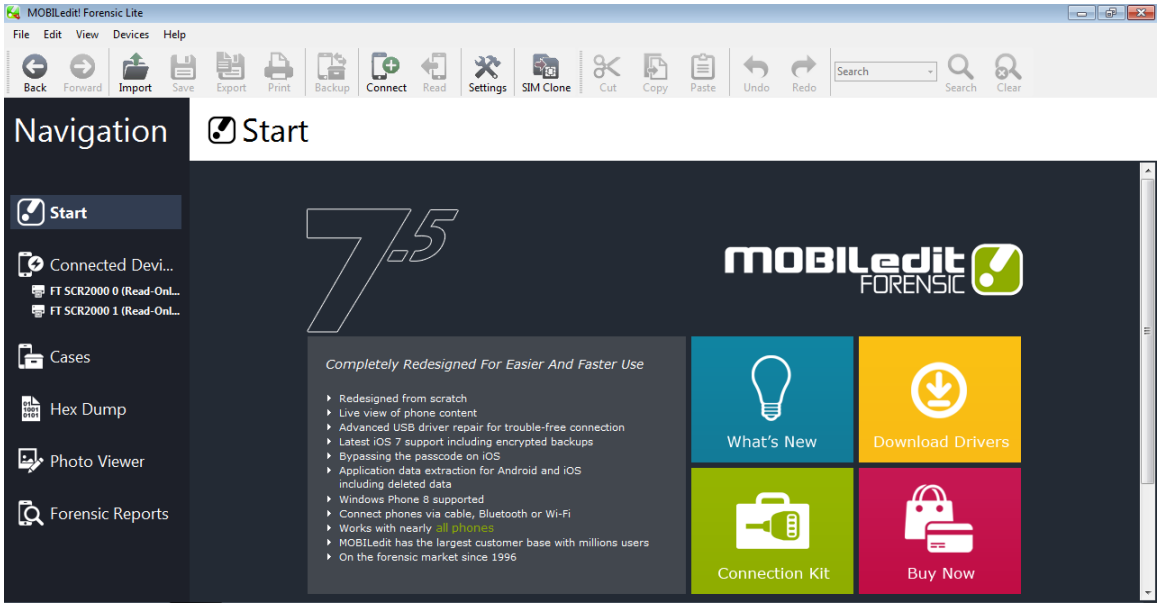
Obrázek B.5: Výběr dat

Po dokončení extrakce jsou získaná data zobrazena. Data je možné exportovat do pdf.

B.2 MOBILedit! Forensic

Instalace a spuštění

Program je určený pro OS Windows, po nainstalování ho spusťte ikonou na ploše.

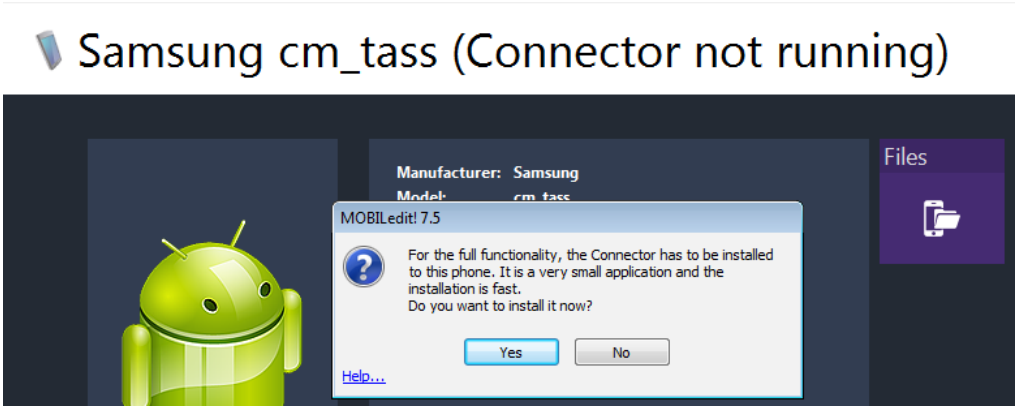


Obrázek B.6: Úvodní obrazovka programu MOBILedit!

Přidání zkoumaného zařízení

Po spuštění programu připojte zkoumané zařízení. Pokud jej program automaticky nerozpozná, spusťte průvodce připojením klepnutím na ikonu *Connect*.

Po rozpoznání připojeného zařízení je nabídnuta instalace aplikace (*Connector*). Potvrďte ji. Aplikace se po nainstalování spustí a na displeji zařízení zobrazuje stav spojení, popřípadě indikuje jinou (právě probíhající) činnost.

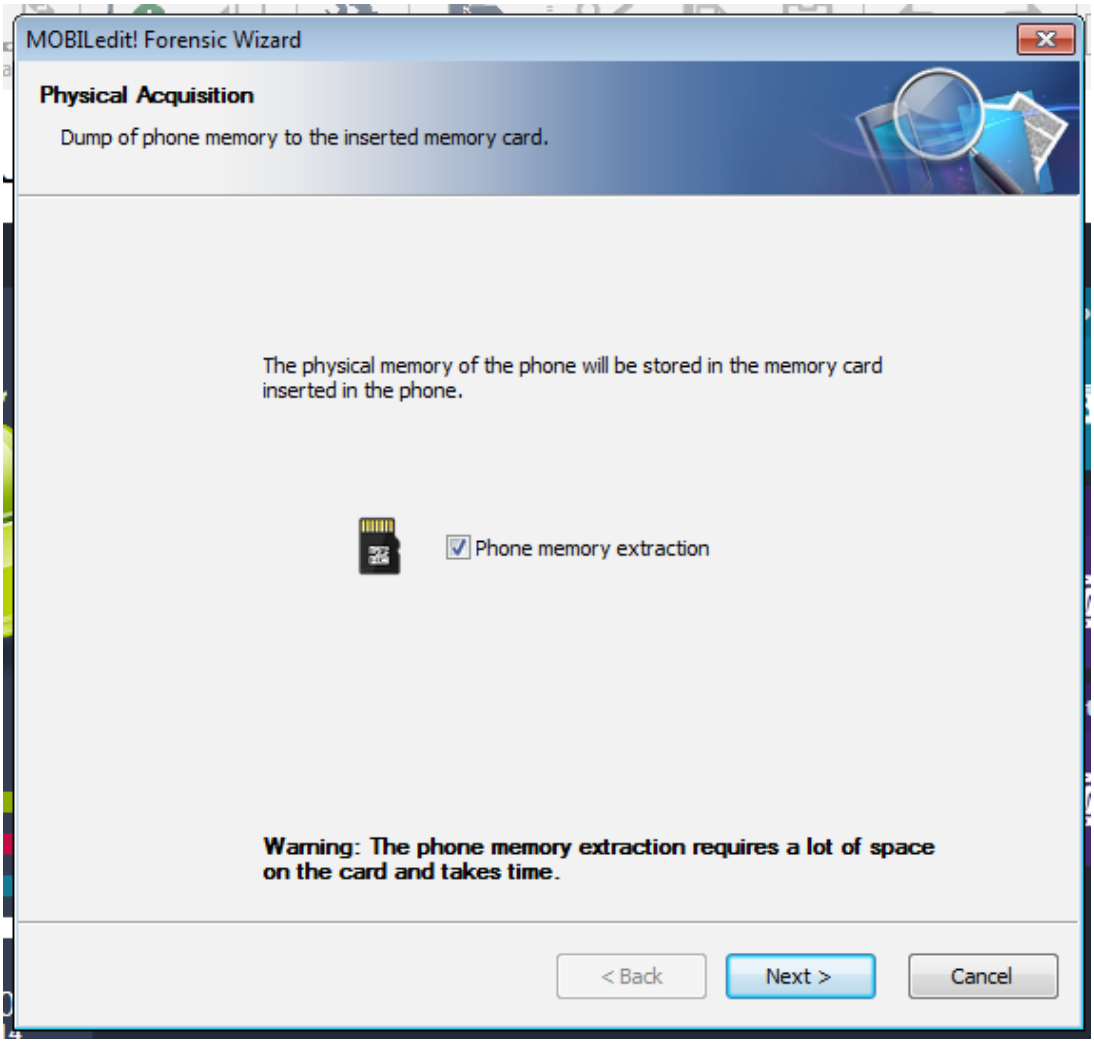


Obrázek B.7: Instalace aplikace do připojeného zařízení

Extrakce dat

Dalším krokem je načtení dat z telefonu. Stiskněte tlačítko *Backup*. Spustí se průvodce, v tom je možné zvolit konkrétní položky, které chceme ze zařízení získat.

Po získání vybraných dat je uživateli nabídnuto uložení obrazu paměti zařízení na paměťovou kartu.



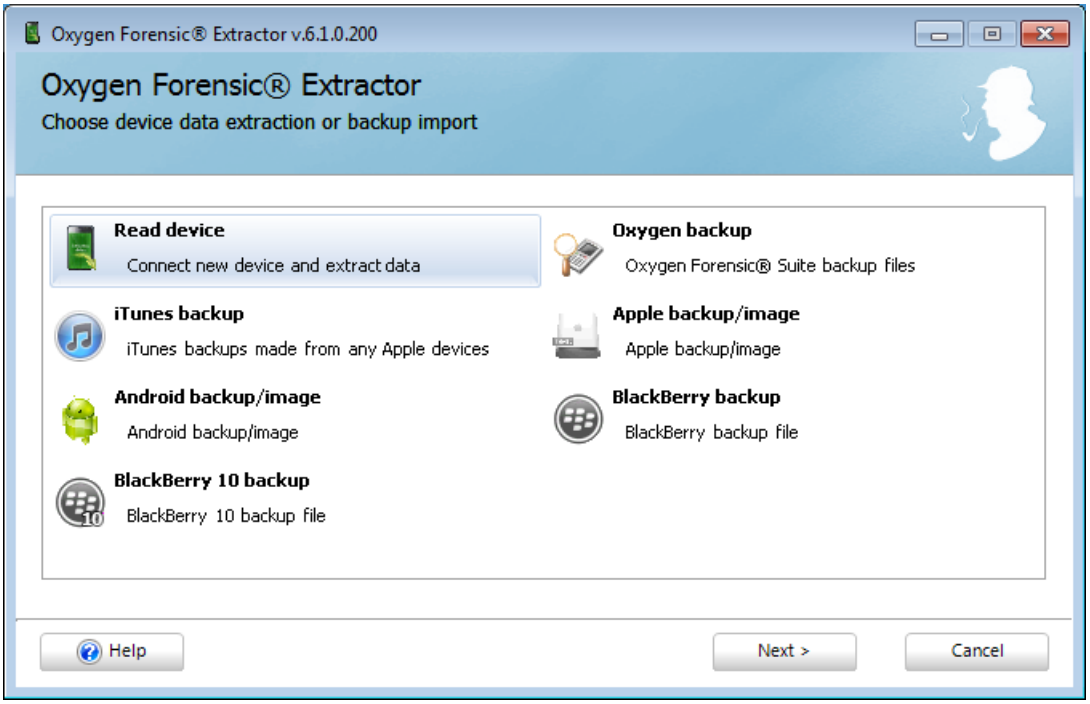
Obrázek B.8: Uložení obrazu paměti

Po dokončení extrakce je nabídnuta možnost vytvořit zprávu o získaných datech v několika formátech/jazycích. Ve zkušební verzi ale zprávy vytvářet není umožněno. Ukončete průvodce. V programu MOBILedit! jsou nyní k dispozici data z připojeného telefonu a je možné je procházet.

B.3 Oxygen Forensic® Suite

Instalace a spuštění

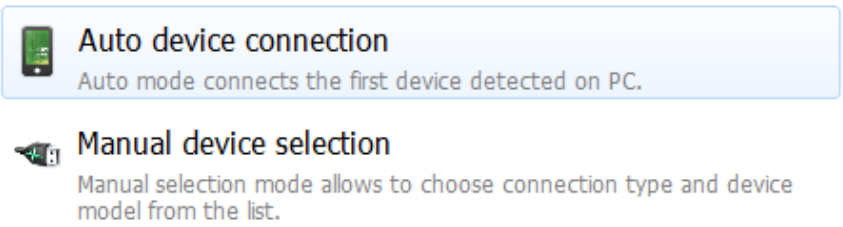
Program je určený pro OS Windows, po nainstalování se na ploše objeví 2 ikony: *Oxygen Forensic Suite* a *Oxygen Data Extraction Wizzard*. Pro přidání nového zařízení poslouží druhá z nich – průvodce extrahováním dat.



Obrázek B.9: Oxygen Data Extraction Wizzard

Přidání zkoumaného zařízení

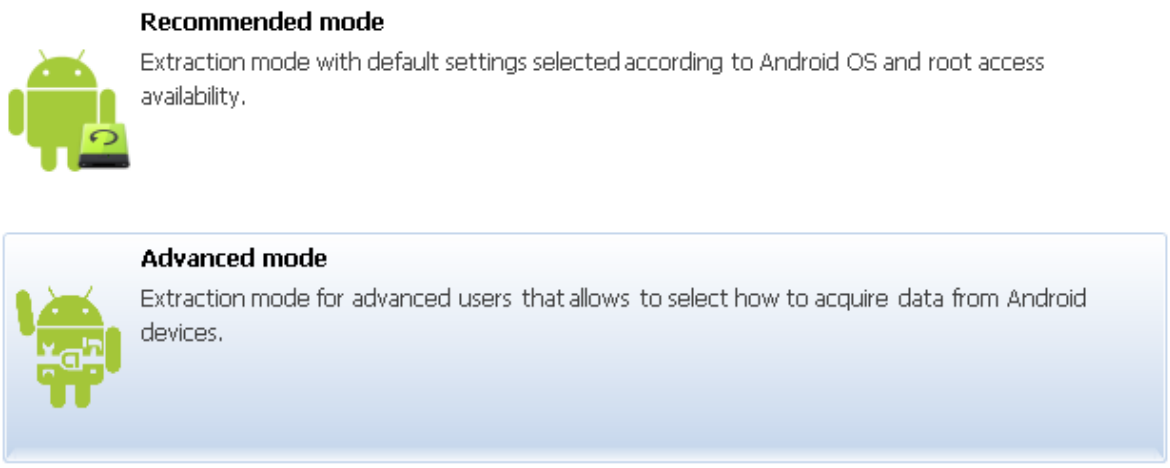
Po spuštění průvodce připojte zkoumané zařízení a zvolte položku *Read device*. Na následující obrazovce lze zvolit metodu rozpoznání zařízení. Pokud by selhala automatická detekce, lze zařízení ručně vybrat ze seznamu.



Obrázek B.10: Volba způsobu rozpoznání zařízení

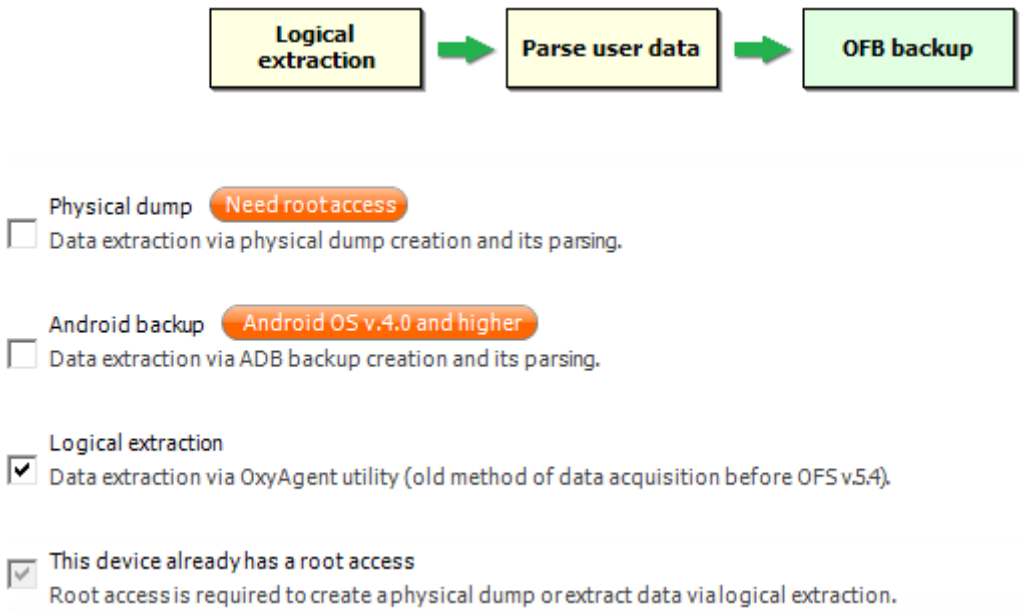
Extrakce dat

Dalším krokem je volba způsobu extrakce dat. Opět je na výběr automatická varianta a pokročilejší ruční varianta, ve které můžeme zvolit použitou metodu. Vybereme *Advanced mode*.



Obrázek B.11: Volba způsobu extrace dat

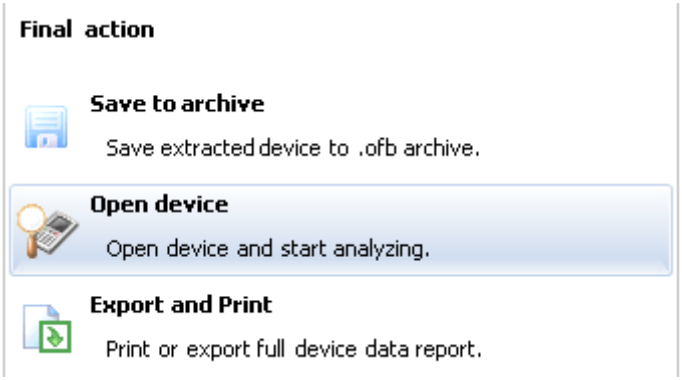
Na další obrazovce se volí použitá metoda. Vybereme *Logical extraction*, která do zařízení nainstaluje aplikaci umožňující propojení s počítačem.



Obrázek B.12: Volba metody

Pokud zařízení nemá proveden *root*, je možné tak učinit zde zatržením poslední volby.

Po dokončení extrakce je nabídnuta možnost vytvoření zprávy o získaných datach, popřípadě požnost otevřít zkoumané zařízení v programu Oxygen Forensic Suite a zde data analyzovat. Ve zkušební verzi jsou však možnosti analýzy omezené. Zvolte proto možnost *Export and Print* pro uložení zprávy do souboru `.pdf` a ukončete průvodce.



Obrázek B.13: Uložení/zobrazení dat